NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE DEVELOPMENT OF A COMPUTER CODE
(U2DIIF)
FOR THE NUMERICAL SOLUTION OF
UNSTEADY, INVISCID AND INCOMPRESSIBLE
FLOW OVER AN AIRFOIL

by

Ngai-Huat Teng

June 1987

Thesis Advisor M. F. Platzer

Approved for public release; distribution is unlimited.

URITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
DECLASSIFICATION / DOWNGRADING SCHEDULE	
PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION val Postgraduate School — 6b OFFICE SYMBOL (If applicable) Code 67	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
NAME OF FUNDING/SPONSORING ORGANIZATION — 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

TITLE (Include Security Classification)
THE DEVELOPMENT OF A COMPUTER CODE (U2DIIF) FOR THE NUMERICAL SOLUTION OF UNSTEADY, INVISCID AND INCOMPRESSIBLE FLOW OVER AN AIRFOIL

PERSONAL AUTHOR(S)
Teng, Ngai-Huat

TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1987, June	136

SUPPLEMENTARY NOTATION

COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computational Fluid Dynamics, Panel Methods, Unsteady Aerodynamics, Potential Flows

ABSTRACT (Continue on reverse if necessary and identify by block number)

A numerical technique is formulated, in a computer program U2DIIF, for the solution of flow over an airfoil executing an arbitrary unsteady motion in an inviscid and incompressible medium. The technique extends the well known Panel Methods for steady flow into solving a non-linear unsteady flow problem arising from the continuous vortex shedding into the trailing wake due to the unsteady motion of the airfoil. Numerous case-runs are presented to verify U2DIIF computer code against other theoretical and/or numerical methods as well as in cases where limited experimental data are obtainable in literatures. These case-runs include airfoils undergoing a step change or a modified ramp change of angle-of-attack, airfoils executing harmonic oscillation in pitching and plunging motions and airfoils penetrating a sharp edge gust.

DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
NAME OF RESPONSIBLE INDIVIDUAL Professor M. F. Platzer	22b TELEPHONE (Include Area Code) (408) 646-2311 — 22c OFFICE SYMBOL Code 67Pl

ORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

The Development of a Computer Code (U2DIIF) for the Numerical Solution of Unsteady, Inviscid and Incompressible Flow Over An Airfoil

by

Ngai-Huat Teng
Ministry of Defence, Republic of Singapore
B.A.(Honours), University of Oxford, United Kingdom, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1987

ABSTRACT

A numerical technique is formulated, in a computer program U2DIIF, for the solution of flow over an airfoil executing an arbitrary unsteady motion in an inviscid and incompressible medium. The technique extends the well known Panel Methods for steady flow into solving a non-linear unsteady flow problem arising from the continuous vortex shedding into the trailing wake due to the unsteady motion of the airfoil. Numerous case-runs are presented to verify U2DIIF computer code against other theoretical and/or numerical methods as well as in cases where limited experimental data are obtainable in literatures. These case-runs include airfoils undergoing a step change or a modified ramp change of angle-of-attack, airfoils executing harmonic oscillation in pitching and plunging motions and airfoils penetrating a sharp edge gust.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF SYMBOLS

A	singularity-type indicator for uniformly distributed source
B	singularity-type indicator for uniformly distributed vorticity
C	singularity-type indicator for concentrated point vortex
c	chord length
C_d	2-dimensional drag coefficient
C_ℓ	2-dimensional lift coefficient
$C_{\ell\infty}$	steady state value of C_ℓ
C_m	2-dimensional pitching moment coefficient about the leading edge
$C_{m\infty}$	steady state value of C_m
C_p	pressure coefficient
C_x	x-force coefficient
C_y	y-force coefficient
h_x	chordwise translational position (positive forward)
h_y	transverse translational position (positive downward)
ℓ	perimeter length of airfoil
M	Mach number
m	number of concentrated core-vortices
n	total number of panels
i, **j**	unit vectors directed along the x- and y-directions
n, **t**	unit vectors normal and tangential to panel
P	static pressure
P_∞	freestream static pressure
q	dimensionless strength of uniformly distributed panel source
r	scalar distance between 2 points indicated by its indices
s	distance taken clockwise along the airfoil contour
t	time step
$\mathbf{V}_\infty$	freestream velocity vector
V_∞	magnitude of the freestream velocity (scalar)
V^n	total velocity component normal to panel
V^t	total velocity component tangential to panel
U, V	absolute velocity components resolved in the x- and y-directions

(x,y)	coordinate system fixed on the airfoil
(xm,ym)	mid point of panel (control point coordinates)
z	total number of panels ahead of airfoil (calculation of φ)
α, AOA	angle of attack (positive clockwise from V_∞)
α_i	initial angle of attack
β	geometric angle in radian used in computing influence coefficients
Γ	dimensionless circulation strength (positive clockwise)
γ	dimensionless strength of uniformly distributed panel vorticity
Δ	length of the shed vorticity panel
$\delta\alpha$	change in AOA from α_i or amplitude of pitch oscillation
δh_x, δh_y	amplitudes of chordwise and transverse oscillations
Θ	orientation angle of the shed vorticity panel with x-direction
θ	inclination angle of panel to the x-axis
	angle made with x-axis by the line joining a field point to a point singularity
λ	phase difference of the chordwise- from the transverse-oscillation
ρ	incompressible density
τ	dimensionless rise time for the ramp change in AOA
Φ	total velocity potential
φ_∞	velocity potential due to freestream
φ	velocity potential due to disturbances
φ_s	velocity potential due to source distributions
φ_v	velocity potential due to vorticity distributions
φ_{cv}	velocity potential due to core vortices in the wake
Ω	pitch angular rate of airfoil (positive counterclockwise)
ω	harmonic oscillation frequency

Superscript Indices :

n	normal component
t	tangential component
x	x-component
y	y-component

Subscript Indices :

0	indicator for $t=0$
i, j	indicators for the airfoil panels and nodes
k	indicator for time step

le	leading edge indicator
m, h	indicators for wake core vortices
f	indicator for panels and nodes ahead of airfoil (calculation of φ)
w	indicator for the shed vorticity panel
φ	contribution due to disturbance potential

Operators :

∂	partial derivative
∇	gradient
$\int$	integral
$\sum$	summation
$\sqrt{}$	square-root

ACKNOWLEDGEMENTS

My sincere appreciation goes to Professor M.F. Platzer for his professional guidance in the conduct of this thesis research that provided me with the opportunity to work in the area of Computational Fluid Dynamics which interested me most in my entire course of MSc studies in the Naval Postgraduate School.

I would also like to express my gratitude to Mr. A. Krainer for his numerous suggestions and constructive criticism, without which fewer case-runs for the validation of U2DIIF code as seen in this documentation would have been produced.

A word of thanks is also due to my wife, Florence, for her encouragement and support during the entire period of this thesis research work.

I. INTRODUCTION

A. GENERAL

In this thesis, a numerical method is formulated and coded in a FORTRAN computer program, codename U2DIIF (Unsteady 2-Dimensional Inviscid Incompressible Flow), to solve for the flow over an airfoil which is executing an unsteady time-dependent motion in an inviscid, incompressible medium.

B. APPROACH

The basic approach to this problem is the extension of a very general and powerful technique, called *Panel Methods*, developed by Hess & Smith [Ref. 1] for steady potential flow problems, to include the unsteady motion of the airfoil that is continuously shedding vorticity into the trailing wake. This vortex shedding process creates the non-linearity effects of the problem in that the wake vortices influence the flow over the airfoil which in turn alters the vortex shedding as the airfoil proceeds in time. It is this very non-linearity of unsteady flow that distinguishes itself from the well known steady Panel Methods solution where the mathematical formulation of the problem results in a set of N linear equations in N unknowns which are solved easily with the standard Gaussian elimination algorithm.

The unsteady flow problem is, however, deprived of this relatively easy solution technique. Instead, an iterative type of solution is needed for this non-linear problem. The correct mathematical model must therefore be formulated to describe the vortex shedding process that provides the *mechanism* for the iteration to proceed towards a converged set of solution in each time step.

It is the objective of this thesis to develop a numerical computer program that performs this non-linear potential flow calculation which proceeds step by step in time. At each time step, a complete set of potential flow solutions, inclusive of the airfoil pressure distribution, force and moment coefficients, and the trailing vortex wake pattern (strengths and positions of shed vortices), is obtained.

C. SCOPE

The Panel Method of Hess & Smith, which utilises both the distributed sources and vorticities as panel singularities, for steady flow solution is described in Chapter II.

Chapter III formulates the mathematical model for the unsteady flow problem and its solution procedures, highlighting the essential features in solving the non-linear problem of unsteady flow.

Chapter IV describes the computer program U2DIIF, its essential capabilities, limitations and the necessary input set-up for typical case-runs.

The results of some of the case-runs are presented in Chapter V. They are compared with other theoretical and/or numerical methods as well as in cases where limited experimental data are obtainable in the literature. These case-runs include airfoils undergoing a step change or a modified ramp change in angle-of-attack, airfoils executing harmonic oscillation in both pitching and plunging motions and airfoils penetrating a sharp edge gust.

In the concluding remarks of Chapter VI, the future development and application potential of this numerical method to other studies of unsteady 2-dimensional inviscid incompressible flow are mentioned.

II. STEADY FLOW PROBLEM FORMULATION

A. FRAME OF REFERENCE

Consider a 2-dimensional airfoil in motion with constant linear velocity $-V_\infty$ as shown in Figure 2.1. Using an (x,y) coordinate system fixed on the airfoil, where the x-axis coincides with the chord line originating from the leading edge towards the trailing edge of the airfoil, the flow in this frame of reference is *steady*. That is to say, the fluid velocity and pressure in the flow field depend only on the spatial coordinates (x,y) and not on time. The airfoil then appears to be submerged in an onset flow whose velocity is V_∞ and making an angle of attack, α, with the x-axis (see Figure 2.1).

B. STEADY FLOW PANEL METHODS

1. Definition of Nodes and Panels

The airfoil surface is divided into (n) straight-line segments, called *panels*, by (n+1) arbitrary chosen points, called *nodes*, distributed over the airfoil contour as shown in Figure 2.2. The panel numbering sequence starts with panel 1 on the lower surface at the airfoil trailing edge and proceeds clockwise around the airfoil contour so that the last panel (panel n) ends on the upper surface, also at the airfoil trailing edge.

Notice that this numbering sequence dictates that the airfoil body always lies on the right hand side of the i^{th} panel as one proceeds from the i^{th} node to the $(i+1)^{th}$ node. Also the 1^{st} and the $(n+1)^{th}$ nodes coincide at the trailing edge. It therefore facilitates, as shown in Figure 2.2, the common definition of unit normal vector $\mathbf{n}_i$ and the unit tangential vector $\mathbf{t}_i$ for all panels, i.e., $\mathbf{n}_i$ is directed outward from the body into the flow and $\mathbf{t}_i$ is directed from the i^{th} node to the $(i+1)^{th}$ node.

2. Distribution of Singularities

Figure 2.2 also indicates that a uniform source distribution q_j and a uniform vorticity distribution γ are placed on the j^{th} panel. The source strength q_j varies from panel to panel whereas the vorticity strength γ remains the same for all panels. This particular choice of singularity distributions is one of the many types of singularity combinations (it happened to be the pioneering one though) ever used in a wide variety of the so called *Panel Methods.* The success of representing the flow past an arbitrary shaped airfoil by surface singularity distributions lies in the fact that these singularity distributions automatically satisfy Laplace's equation, the governing flow equation for

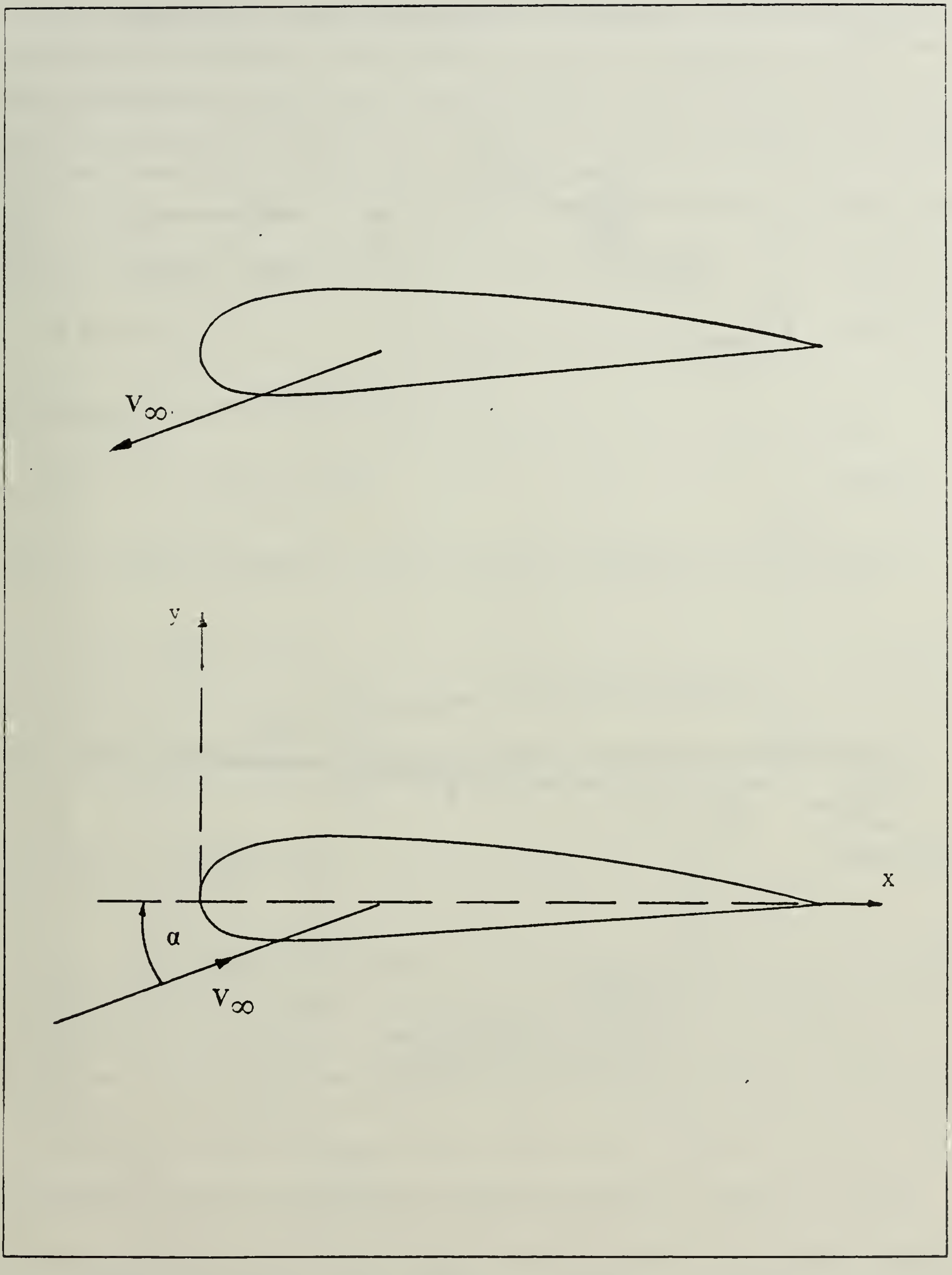


Figure 2.1 Frame of Reference for Steady Flow.

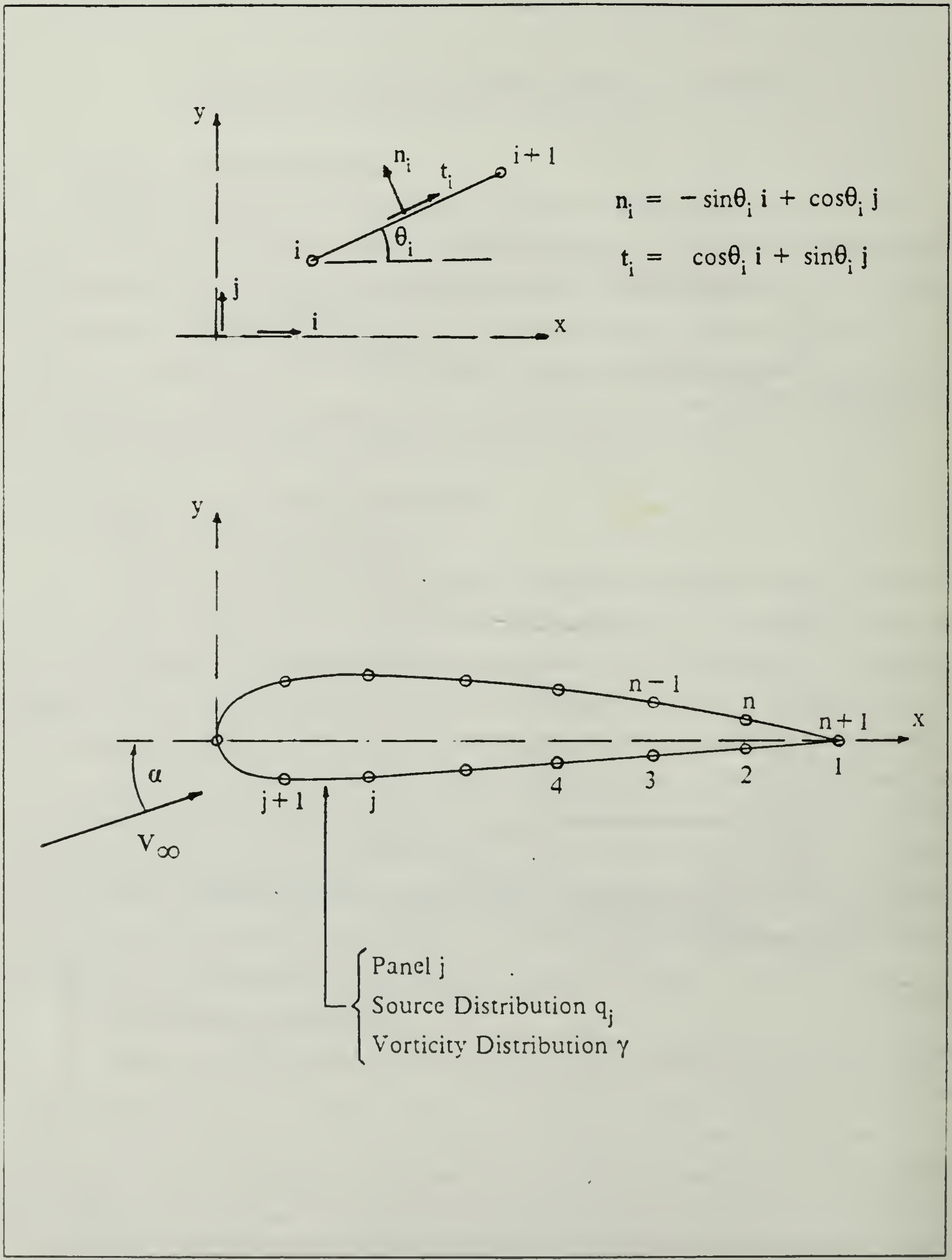


Figure 2.2 Panel Methods Representation for Steady Flow.

inviscid incompressible flow, and the boundary condition at the far field (∞). In addition, the superposition principle applies to any linear homogeneous second order partial differential equation such as Laplace's equation. Therefore one can build up an overall complicated flow field by the combination of simple flows if the appropriate boundary conditions on the airfoil can be satisfied accurately. For our case the overall flow field (represented by the velocity potential Φ) can be built up by three simple flows,

$$\Phi = \varphi_{\infty} + \varphi_s + \varphi_v \qquad \text{(eqn 2.1)}$$

where φ_{∞} is the potential of the onset flow,

$$\varphi_{\infty} = V_{\infty} (x \cos\alpha + y \sin\alpha) \qquad \text{(eqn 2.2)}$$

φ_s is the velocity potential of a source distribution of strength q(s) per unit length,

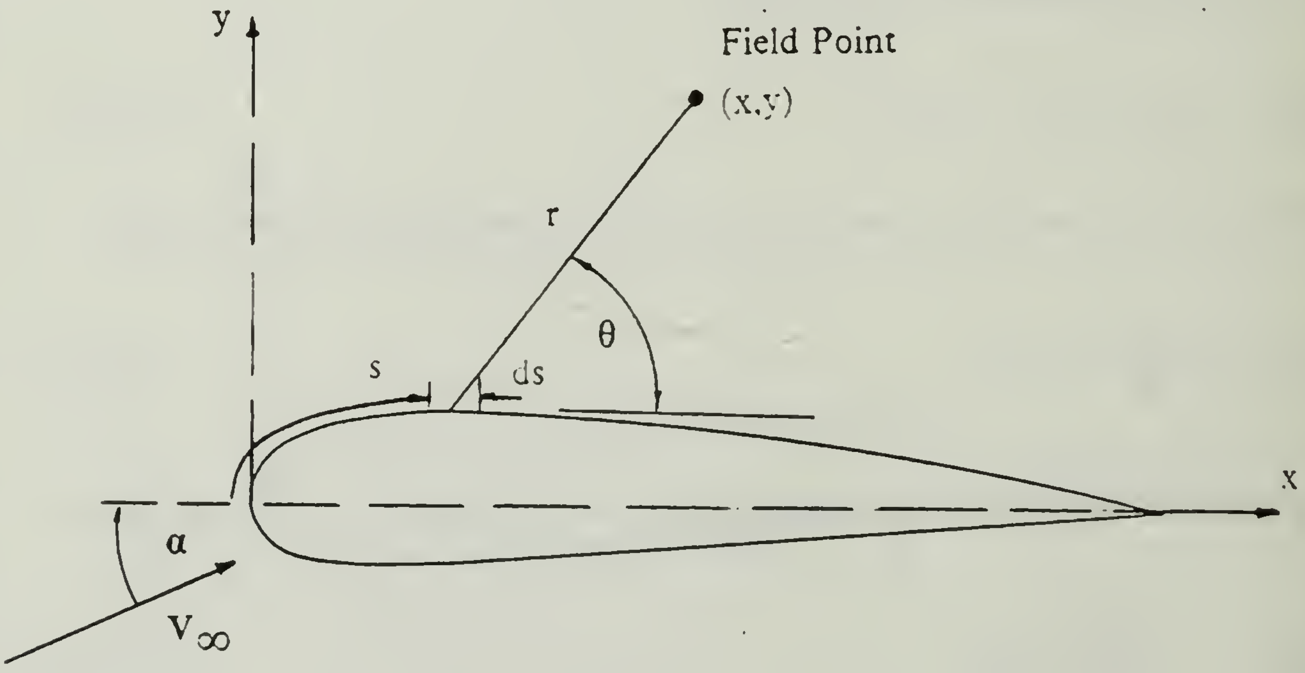
$$\varphi_s = \int \frac{q(s)}{2\pi} \ln r \, ds \qquad \text{(eqn 2.3)}$$

φ_v is the velocity potential of a vorticity distribution of strength $\gamma(s)$ per unit length.

$$\varphi_s = - \int \frac{\gamma(s)}{2\pi} \theta \, ds \qquad \text{(eqn 2.4)}$$

The integrals in Equations 2.3 and 2.4 are performed along the surface contour s and (r,θ) are polar coordinates of any field point (x,y) measured from the airfoil surface at an arbitrary point as shown in Figure 2.3. The difficult task of evaluating these integrals has been greatly simplified by our singularity distributions postulated to represent the flow over the airfoil: that is, instead of integrating over the entire airfoil contour, we integrate on each panel along a straight line where q_j and γ are constant, then sum up the effects of all panels. Equation 2.1 therefore becomes,

$$\Phi = V_{\infty} (x \cos\alpha + y \sin\alpha) + \sum_{j=1}^{n} \int_{\text{panel } j} \left[\frac{q_j}{2\pi} \ln r - \frac{\gamma}{2\pi} \theta \right] ds \qquad \text{(eqn 2.5)}$$



$$\Phi = V_{\infty} \left(x \cos\alpha + y \sin\alpha \right) + \int \frac{q(s)}{2\pi} \ln r \, ds - \int \frac{\gamma(s)}{2\pi} \theta \, ds$$

Figure 2.3 Potential Evaluation at a Field Point.

It can be seen from Equation 2.5 that Φ is completely defined if the $(n+1)$ unknowns, q_j $(j=1,2,\ldots,n)$ and γ, can be calculated using a numerical technique yet to be described. Once the potential Φ is solved, the velocity can be evaluated by taking *grad* Φ. At this point we introduce a definition of *disturbance potential*, φ, as the sum of potential due to both the source and vorticity distribution,

$$\varphi = \varphi_s + \varphi_v \qquad \text{(eqn 2.6)}$$

Equation 2.1 therefore reads,

$$\Phi = \varphi_\infty + \varphi \qquad \text{(eqn 2.7)}$$

The total velocity vector is thus,

$$\begin{aligned} V_{total} &= \nabla\Phi \\ &= \nabla\varphi_\infty + \nabla\varphi \\ &= V_\infty + \nabla\varphi \qquad \text{(eqn 2.8)} \end{aligned}$$

The pressure can be obtained from Bernoulli's Equation,

$$C_p = \frac{P - P_\infty}{\frac{1}{2}\rho V_\infty^2} = 1 - \left(\frac{V_{total}}{V_\infty}\right)^2 \qquad \text{(eqn 2.9)}$$

Notice that Figure 2.3 indicates that the field point lies off the airfoil surface, however, we are interested in field points that are on the airfoil surface. In the case of steady flow, the expressions for V_{total} and C_p are the same for field points lying on or off the airfoil surface. It is nevertheless not the same in unsteady flow, as will be seen in Chapter III, in that V_{total} must include the rigid body motion of the airfoil when one evaluates field points on the airfoil surface.

3. **Boundary Conditions**

The boundary conditions to be satisfied include the flow tangency conditions and the Kutta Condition. The flow tangency conditions are satisfied at the exterior mid points, called *control points*, of all panels by taking the resultant velocity at each control point to have only $(V^t)_i$ but,

$$(V^n)_i = 0 , \quad i = 1,2,....,n \qquad \text{(eqn 2.10)}$$

where $(V^t)_i$ and $(V^n)_i$ are the tangential and normal components of the total velocity at the control point of the i^{th} panel due to the free stream and the velocities induced by the source and vorticity distributions on all the panels, j (j = 1,2,....,n).

The Kutta condition postulates that the pressures on the upper and lower panels at the trailing edge be equal in order that the flow leaves the trailing edge smoothly. By using Bernoulli's equation for steady potential flow, this pressure equilibrium condition implies that the tangential velocities in the downstream direction at the 1^{st} and the n^{th} panel control points must be equal. This fact is certainly consistent with the knowledge that when steady flow is established, the total circulation over the airfoil does not change if the tangential velocities are the same at the trailing edge panels.

$$(V^t)_1 = -(V^t)_n \qquad \text{(eqn 2.11)}$$

If one could explicitly express Equations 2.10 and 2.11 in terms of the unknowns q_j (j = 1,2,....,n) and γ, the task is then reduced to solving a linear system of (n + 1) simultaneous equations for the (n + 1) unknowns.

C. INFLUENCE COEFFICIENTS

1. The Concept of Influence Coefficients

The numerical technique employed in Panel Methods to manipulate equations 2.10 and 2.11 into an algebraic system of linear simultaneous equations involves the important concept of *influence coefficients.* An influence coefficient is defined as the velocity induced at a field point by a unit strength singularity (be it a point singularity or a distributed singularity) placed anywhere within the flow field. In this case, it is the unit strength singularity distribution on one panel. Recall that equations 2.10 and 2.11 simply require the computation of the normal and tangential velocity components at all the panel control points. The normal components of velocities are essential in satisfying flow tangency conditions while the tangential components of velocities are necessary for satisfying the Kutta condition as well as computing the pressure distribution. The procedure is thus to compute, at the i^{th} panel control point, the velocity components induced by the source and vorticity distributions on all the panels, j (j = 1,2,....,n), including the i^{th} panel itself. Summation

of all the induced velocities, separately for the normal and tangential components, together with the free stream velocity components produces all the required $(V^n)_i$ and $(V^t)_i$, $i = 1,2,....,n$.

2. **Notation for Influence Coefficient**

We shall adopt a consistent set of notation for the influence coefficients used throughout this documentation. It is so designated to permit easy recognition in that each influence coefficient contains all the associated information one needs. An influence coefficient is denoted with a superscript and two subsript as follows:

$$\chi^s{}_{pq}$$

where χ denotes the type of singularity involved, we shall arbitrarily use A, B and C for the uniformly distributed source, uniformly distributed vorticity and point vortex respectively. The superscript s is an indicator telling which component the induced velocity is. The first subscript p identifies the field point where the induced velocity is evaluated. The second subscript q denotes the particular singularity contributing to the induced velocity.

We thus define, for the steady flow problem, the following influence coefficients :

- $A^n{}_{ij}$: normal velocity component induced at the i^{th} panel control point by unit strength source distribution on the j^{th} panel.
- $A^t{}_{ij}$: tangential velocity component induced at the i^{th} panel control point by unit strength source distribution on the j^{th} panel.
- $B^n{}_{ij}$: normal velocity component induced at the i^{th} panel control point by unit strength vorticity distribution on the j^{th} panel.
- $B^t{}_{ij}$: tangential velocity component induced at the i^{th} panel control point by unit strength vorticity distribution on the j^{th} panel.

3. **Computation of Influence Coefficients**

The influence coefficients turn out to be related, not surprisingly, to the geometry of the airfoil and the manner in which the panels are formed. Specifically, as derived in [Ref. 2], the A's and B's influence coefficients,[1] due to uniformly distributed source or vorticity are functions of :

- The natural logarithm of the ratio of distance from the i^{th} panel control point (the field point) to the $(j+1)^{th}$ and j^{th} nodes of the j^{th} panel where singularities are distributed.

[1] C's coefficients will be needed only for unsteady flow.

- The angle, in radian, subtended at the i^{th} panel control point (the field point) by the $(j+1)^{th}$ and j^{th} nodes of the j^{th} panel where singularities are distributed.
- The trigonometry angles of the i^{th} and j^{th} panels.

Referring to the geometrical quantities indicated in Figure 2.4, the expressions[2] for these influence coefficients are :

$$A^n_{ij} = \frac{1}{2\pi}\left[\sin(\theta_i-\theta_j)\ln\frac{r_{i,j+1}}{r_{ij}} + \cos(\theta_i-\theta_j)\,\beta_{ij}\right], \quad \text{for } i \neq j$$
$$= \frac{1}{2}, \quad \text{for } i = j \qquad \text{(eqn 2.12)}$$

$$A^t_{ij} = \frac{1}{2\pi}\left[\sin(\theta_i-\theta_j)\,\beta_{ij} - \cos(\theta_i-\theta_j)\ln\frac{r_{i,j+1}}{r_{ij}}\right], \quad \text{for } i \neq j$$
$$= 0, \quad \text{for } i = j \qquad \text{(eqn 2.13)}$$

$$B^n_{ij} = \frac{1}{2\pi}\left[\cos(\theta_i-\theta_j)\ln\frac{r_{i,j+1}}{r_{ij}} - \sin(\theta_i-\theta_j)\,\beta_{ij}\right], \quad \text{for } i \neq j$$
$$= 0, \quad \text{for } i = j \qquad \text{(eqn 2.14)}$$

$$B^t_{ij} = \frac{1}{2\pi}\left[\cos(\theta_i-\theta_j)\,\beta_{ij} + \sin(\theta_i-\theta_j)\ln\frac{r_{i,j+1}}{r_{ij}}\right], \quad \text{for } i \neq j$$
$$= \frac{1}{2}, \quad \text{for } i = j \qquad \text{(eqn 2.15)}$$

where :

$$r_{i,j+1} = \sqrt{[(xm_i - x_{j+1})^2 + (ym_i - y_{j+1})^2]}$$

$$r_{ij} = \sqrt{[(xm_i - x_j)^2 + (ym_i - y_j)^2]}$$

$$xm_i = \tfrac{1}{2}(x_i + x_{i+1})$$

$$ym_i = \tfrac{1}{2}(y_i + y_{i+1})$$

---

[2]Actual computation uses $A^t_{ij} = -B^n_{ij}$ and $B^t_{ij} = A^n_{ij}$ to reduce computing time.

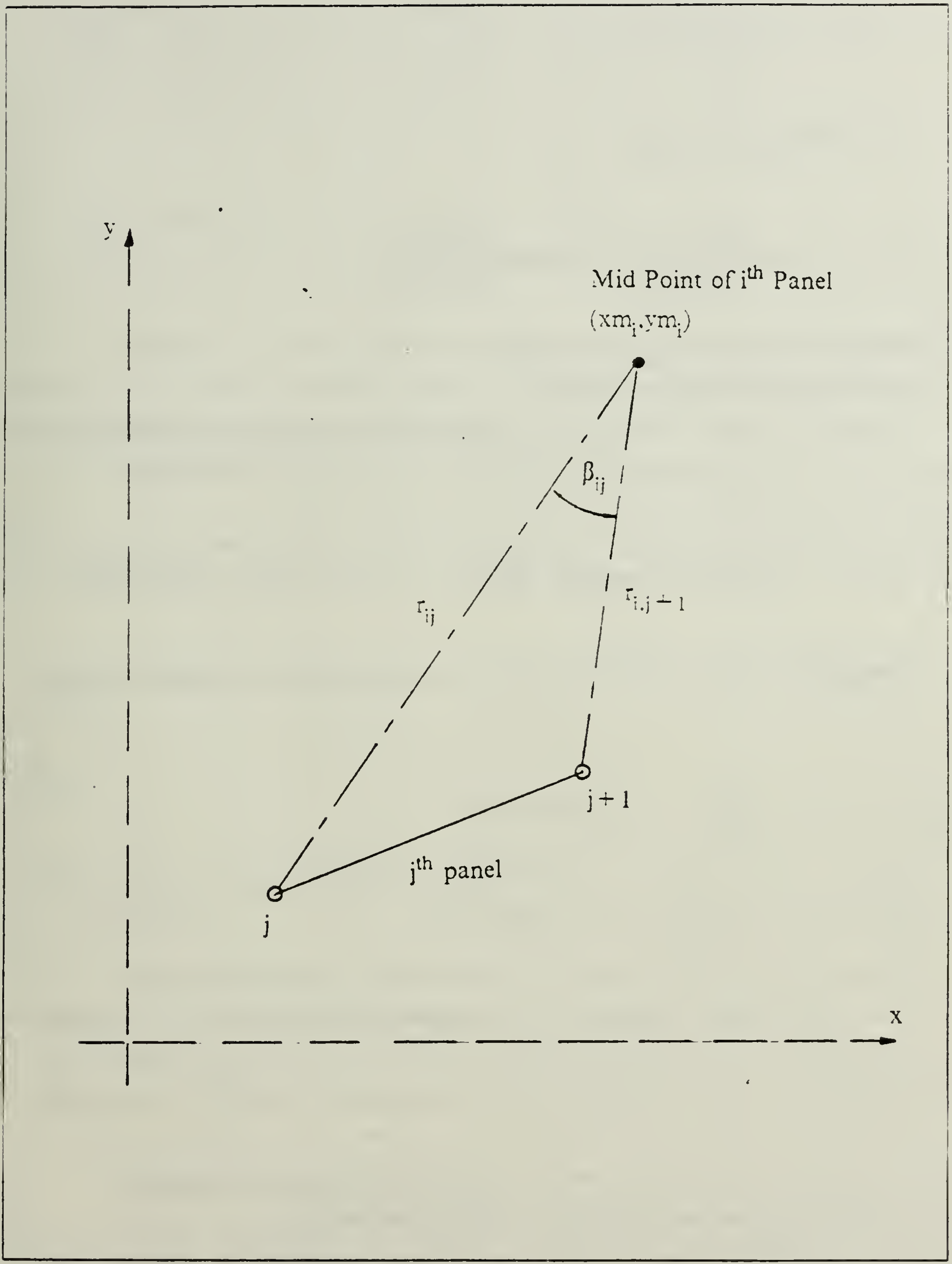


Figure 2.4 Influence Coefficients due to Uniformly Distributed Singularities.

$$\theta_i = \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$$

$$\theta_j = \arctan\left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j}\right)$$

$$\beta_{ij} = \arctan\left(\frac{ym_i - y_{j+1}}{xm_i - x_{j+1}}\right) - \arctan\left(\frac{ym_i - y_j}{xm_i - x_j}\right)$$

D. NUMERICAL SOLUTION SCHEME

1. Rewriting the Boundary Conditions

Using the concept of influence coefficients, the flow tangency conditions of Equation 2.10 can be expressed as,

$$\sum_{j=1}^{n} [\, A^n_{ij}\, q_j \,] + \gamma \sum_{j=1}^{n} B^n_{ij} + V_\infty \sin(\alpha - \theta_i) = 0 \;, \qquad i = 1,2,\ldots.,n \qquad \text{(eqn 2.16)}$$

The Kutta condition of Equation 2.11, in terms of influence coefficients, looks as,

$$- \sum_{j=1}^{n} [\, A^t_{1j}\, q_j \,] - \gamma \sum_{j=1}^{n} B^t_{1j} - V_\infty \cos(\alpha - \theta_1)$$

$$= \sum_{j=1}^{n} [\, A^t_{nj}\, q_j \,] + \gamma \sum_{j=1}^{n} B^t_{nj} + V_\infty \cos(\alpha - \theta_n) \qquad \text{(eqn 2.17)}$$

The negative signs appearing on the left-hand-side of Equation 2.17 are a direct consequence of our definition of unit tangential vector. In other words, the tangential velocities on the lower surface panels downstream of the front stagnation point have negative values. This feature in fact allows one to predict the front stagnation point by interpolating the velocity distribution around the leading edge.

2. Solving the Strengths of Source and Vorticity Distributions

It is not difficult at this stage to see that if we collect the like terms in Equation 2.17 and expand Equation 2.16 for all i's (i = 1,2,....,n), these equations constitute none other than a linear algebraic system of (n + 1) equations as shown in the matrix Equation 2.18.

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n+1} \\
a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n+1} \\
a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n+1} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
a_{n,1} & a_{n,2} & \cdots & \cdots & a_{n,n+1} \\
a_{n+1,1} & \cdots & \cdots & \cdots & a_{n+1,n+1}
\end{bmatrix}
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \cdot \\ \cdot \\ q_n \\ \gamma \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdot \\ \cdot \\ b_n \\ b_{n+1} \end{bmatrix}
\qquad \text{(eqn 2.18)}
$$

Equation 2.18 is a set of linearly independent equations which can be easily solved by any standard linear system solver, one of which is the well known method of *Gaussian Elimination with Partial Pivoting.*

3. Computation of Velocity and Pressure Distribution

Once the q_j ($j=1,2,....,n$) and γ are solved, the velocities at all the panel control points can be evaluated. Only the tangential components exist since the normal components are already set to zeroes by the flow tangency conditions.

$$
\frac{V_{total}}{V_\infty} = (V^t)_i \ , \qquad i=1,2,....,n \qquad \text{(eqn 2.19)}
$$

where :

$$
(V^t)_i = \sum_{j=1}^{n} [\, A^t_{ij}\, q_j \,] + \gamma \sum_{j=1}^{n} B^t_{ij} + V_\infty \cos(\alpha - \theta_i) \ , \qquad i=1,2,....n \qquad \text{(eqn 2.20)}
$$

Substituting Equation 2.20 into the C_p equation (Equation 2.9), the pressure coefficients at the i^{th} panel control point is :

$$
(C_p)_i = 1 - (V^t)_i^2 \ , \qquad i=1,2,....,n \qquad \text{(eqn 2.21)}
$$

4. Computation of Forces and Moments

The two dimensional aerodynamic coefficients of lift (C_ℓ), drag (C_d) and pitching moment (C_m) about the leading edge are computed by integration of the pressure distribution assuming constant C_p exists in each panel. The computation is

first done by integrating forces in the airfoil-fixed coordinate system followed by a rotation to the respective lift and drag directions along and perpendicular to the free stream (V_∞) as follows :

$$C_y = \sum_{i=1}^{n} (C_p)_i \, (x_{i+1} - x_i) \qquad \text{(eqn 2.22)}$$

$$C_x = -\sum_{i=1}^{n} (C_p)_i \, (y_{i+1} - y_i) \qquad \text{(eqn 2.23)}$$

$$C_m = \sum_{i=1}^{n} (C_p)_i \, [(x_{i+1} - x_i)\, xm_i + (y_{i+1} - y_i)\, ym_i\,] \qquad \text{(eqn 2.24)}$$

$$C_d = C_x \cos\alpha + C_y \sin\alpha \qquad \text{(eqn 2.25)}$$

$$C_\ell = C_y \cos\alpha - C_x \sin\alpha \qquad \text{(eqn 2.26)}$$

III. UNSTEADY FLOW PROBLEM FORMULATION

A. OVERVIEW OF UNSTEADY FLOW MODELING

1. Some Previews

Having fully understood the Panel Methods formulation and solution for the steady flow problem, one could then venture into the interesting and complicated unsteady flow case. In this Chapter, we shall see how we could build the time-dependency into the Panel Methods solution which has been proven to be an useful and accurate tool for steady flow. The approach in the unsteady flow problem formulation will proceed, in general, in a manner similar to Chapter II. However, as we go along, we will pick up the highlights of the essential differences (also similarity) between the two problems. Additional flow modeling of the vortex shedding process that greatly influences the numerical solution technique[3] will be discussed in details.

2. Specific Unsteady Flow Model

Recall that in steady flow, the problem is considered solved as soon as the airfoil surface singularity distributions of source and vorticity q_j ($j=1,2,....,n$) and γ are determined. These $(n+1)$ unknowns are, however, time dependent in unsteady flow. We therefore introduce a subscript k as the time-step counter; that is, we postulate to solve the unsteady flow problem at successive intervals of time, starting from $t_0=0$. At each time-step t_k ($k=1,2,....,\infty$), we represent the airfoil by surface singularity distributions consisting of source distribution $(q_j)_k$ ($j=1,2,....,n$) and vorticity distribution γ_k. Again the source strengths vary from panel to panel but the vorticity strength remains the same for all panels.

The overall circulation Γ_k at time-step t_k is simply γ_k multiplied by the airfoil perimeter, ℓ. Since the total circulation in a potential flow field must be preserved according to the *Helmholtz's theorem of continuity of vorticity*, any changes in the circulation on the airfoil surface must be manifested by an equal and opposite change in vorticity in the wake. We call this the *vortex shedding process* and postulate, as shown in Figure 3.1, that this shed vorticity takes place through a small straight line wake element attached as an additional panel to the trailing edge with uniform vorticity distribution $(\gamma_w)_k$. We shall from now on refer to this panel as the *shed vorticity panel*, The shed vorticity panel will be established if its length Δ_k and inclination Θ_k, to x-

[3]Referring to the switch from a *direct* scheme to an *iterative* scheme.

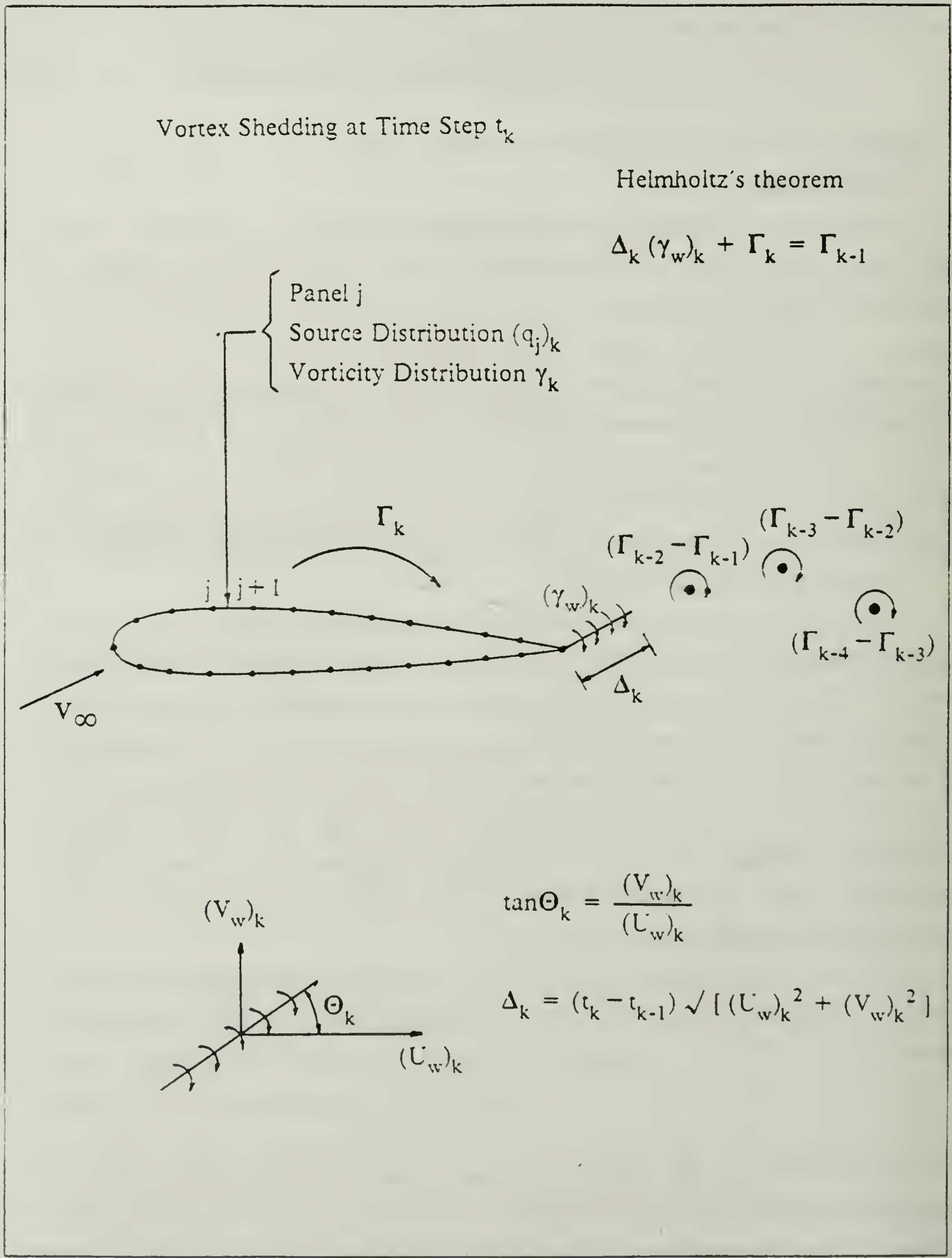


Figure 3.1 Extension of Panel Methods Representation for Unsteady Flow.

axis of the airfoil-fixed coordinate system, satisfy the Helmholtz's theorem as follows,

$$\Delta_k (\gamma_w)_k + \Gamma_k = \Gamma_{k-1} \qquad \text{(eqn 3.1)}$$

or $$\Delta_k (\gamma_w)_k = \Gamma_{k-1} - \Gamma_k = \ell (\gamma_{k-1} - \gamma_k) \qquad \text{(eqn 3.2)}$$

where Γ_{k-1} and γ_{k-1} are respectively the total circulation and vorticity strengths which are already determined at a time-step t_{k-1} before t_k.

Let us project one time step ahead to t_{k+1}, we allow the shed vorticity panel to be detached from the trailing edge and get convected downstream as a concentrated free vortex, with circulation $\Delta_k (\gamma_w)_k$ or $\Gamma_{k-1} - \Gamma_k$, according to the resultant local velocity occurred at the center of the vortex core. At the same time a brand new shed vorticity panel is formed for the new time step and the process is repeated. Therefore the shed vorticity panel model provides exactly the desired communication mechanism to carry the solution from one time-step to another.

We now return back to the time-step t_k and immediately realise that as a result of this continuous vortex shedding, there has been a series of shedding processes occurred prior to t_k that cummulated in a string of concentrated core vortices of strengths $(\Gamma_{k-2} - \Gamma_{k-1})$, $(\Gamma_{k-3} - \Gamma_{k-2})$, $(\Gamma_{k-4} - \Gamma_{k-3})$, ... and so on, forming the wake pattern behind the airfoil as shown in Figure 3.1

The presence of the shed vorticity panel and the downstream resultant wake core vortices do influence the upstream flow in inviscid incompressible flow. In particular the shed vorticity panel itself depends on γ_k to determine its distributed vorticity $(\gamma_w)_k$, this in turn causes changes to $(q_j)_k$ and γ_k. Moreover, the downstream core vortices that constitute the wake are convected under the influence of the free stream and the singularity distributions on the airfoil surface panels including the shed vorticity panel. The problem is thus seen to be coupled from this analytical standpoint. Putting this in simple mathematical terms, the algebriac system of equations (Equation 2.18), representing the flow tangency conditions and Kutta condition for steady flow, are no longer linear because the coefficients a_{ij} in the left-hand-side matrix are not constants anymore. They are function of q_j and γ instead. The presence of non-linearity is indeed what drives the solution scheme into an iterative type for unsteady flow.

3. **Boundary Conditions**

We next investigate whether our unsteady flow model is sufficiently represented, before we could proceed to search for a numerical iterative solution, by matching the unknowns with the available boundary conditions at time-step t_k. Recall that we have introduced three more unknowns $(\gamma_w)_k$, Δ_k and Θ_k in addition to $(q_j)_k$ ($j = 1,2,....,n$) and γ_k. We have, however, so far only identified an extra boundary condition, namely the Helmholtz's theorem (Equation 3.2) in conjunction with the flow tangency conditions at the n panel control points and the Kutta condition of pressure equilibrium at the trailing edge panels. Clearly we are in deficit of two additional conditions before attempting further endeavour to solve the entire system. Basu and Hancock [Ref. 3] suggested the following assumptions :

- The shed vorticity panel is oriented in the direction of the local resultant velocity at the panel mid point.
- The length of the shed vorticity panel is proportional to the magnitude of the resultant velocity at the panel mid point and the step size of the time-step.

Following these assumptions thus permits us to formulate two more boundary conditions as follows,

$$\tan\Theta_k = \frac{(V_w)_k}{(U_w)_k} \qquad \text{(eqn 3.3)}$$

$$\Delta_k = (t_k - t_{k-1}) \sqrt{[\,(U_w)_k^{\,2} + (V_w)_k^{\,2}\,]} \qquad \text{(eqn 3.4)}$$

where $(U_w)_k$ and $(V_w)_k$ are the total velocity components in x and y directions of the airfoil-fixed coordinate system.

The flow tangency conditions are still,

$$[(V^n)_i]_k = 0\ ,\ \ i = 1,2,....,n \qquad \text{(eqn 3.5)}$$

However, the Kutta condition must now include the rates of change of potential at the trailing edge panels (unsteady Bernoulli's equation) which can be related directly to the rate of change of total circulation. By using a backward finite difference approximation for this rate of change of total circulation, we express the Kutta condition as shown in Equation 3.6.

$$[(V^t)_1]_k^2 - [(V^t)_n]_k^2 = 2\left[\frac{\partial}{\partial t}(\varphi_n - \varphi_1)\right]_k = 2\left(\frac{\partial \Gamma}{\partial t}\right)_k$$

$$= 2\,\ell\,\frac{\gamma_k - \gamma_{k-1}}{t_k - t_{k-1}} \qquad \text{(eqn 3.6)}$$

B. RIGID BODY MOTION AND FRAME OF REFERENCE

Consider a rigid airfoil executing a time-dependent motion, comprising linear translation and angular rotation about the leading edge in an inviscid incompressible medium. We can describe this arbitrary motion at any time instant t_k as the vector sum of a mean velocity $-V_\infty$, a time dependent translational velocity $-[U(t)\,\mathbf{i} + V(t)\,\mathbf{j}]$ and a rotational velocity $-\Omega(t)$ where $\mathbf{i}$ & $\mathbf{j}$ are unit vectors in the airfoil-fixed coordinate system as shown in Figure 3.2.

If we continue, as in steady flow, to determine the flow with reference to the (x,y) coordinate system fixed on the airfoil, an observer sitting on this frame of reference sees an unsteady stream velocity, V_{stream}, made up by the vector sum of a mean velocity V_∞, a time dependent translational velocity $[U(t)\,\mathbf{i} + V(t)\,\mathbf{j}]$ and a rotational velocity $\Omega(t)$. Therefore in this frame of reference, unlike the previous case where the airfoil is allowed to move only with constant linear velocity, the flow is still unsteady in that V_{stream} is time dependent.

$$V_{stream} = V_\infty + [(U(t)\,\mathbf{i} + V(t)\,\mathbf{j})] + \Omega(t)\,(y\,\mathbf{i} - x\,\mathbf{j}) \qquad \text{(eqn 3.7)}$$

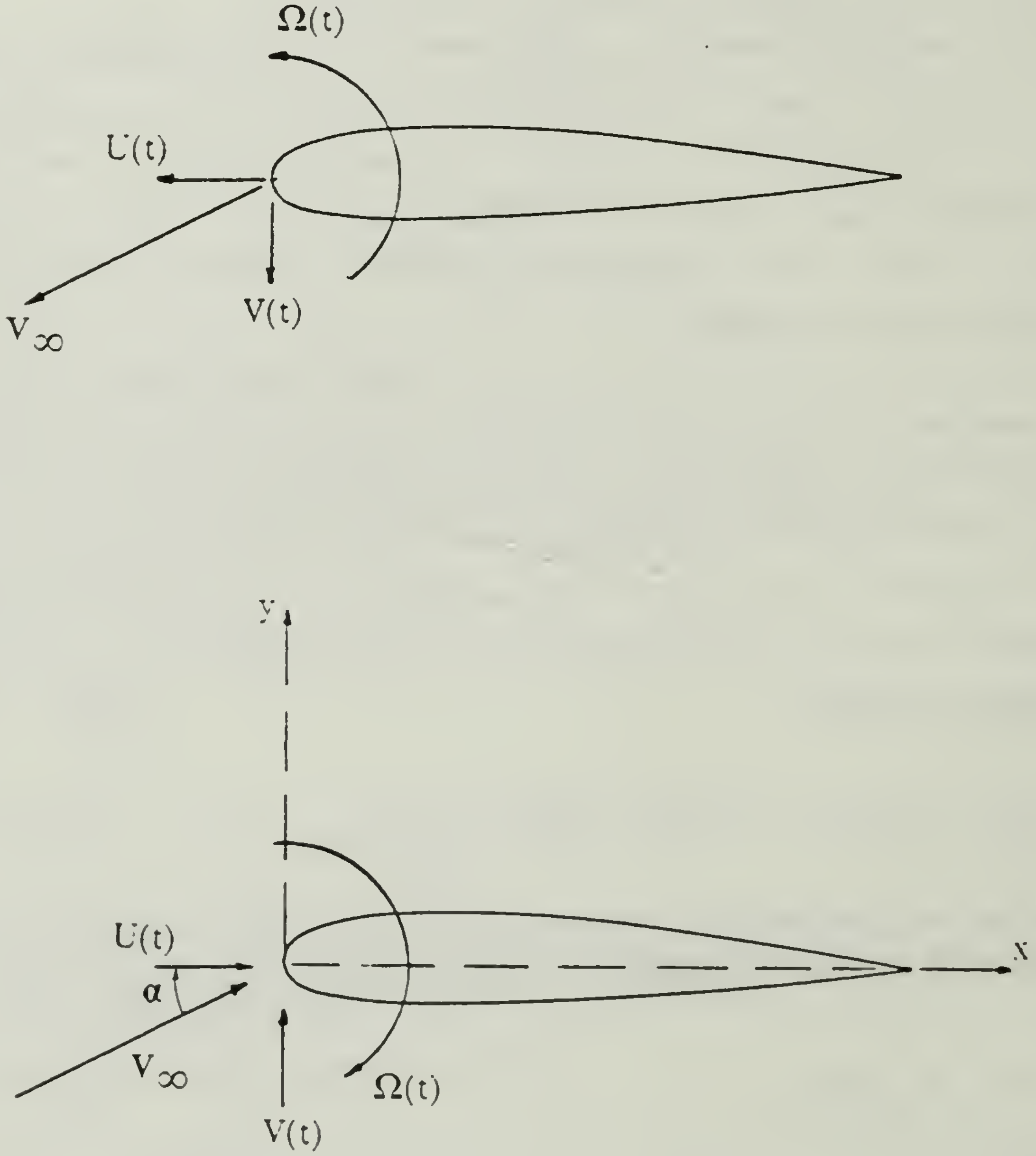
We redefine our disturbance potential to include the potential contributions φ_w and φ_{cv} from the shed vorticity panel and the wake core vortices respectively. Thus :

$$\varphi = \varphi_s + \varphi_v + \varphi_w + \varphi_{cv} \qquad \text{(eqn 3.8)}$$

We then write the total velocity with respect to this frame of reference as,

$$V_{total} = V_{stream} + \nabla\varphi \qquad \text{(eqn 3.9)}$$

Notice that this total velocity is obviously *NOT* the absolute velocity with respect to an inertial coordinate system. Such an inertial coordinate system will be the one where an observer only sees an on-coming flow of V_∞ with constant magnitude and



$$V_{stream} = V_{\infty} - [(U(t)\, i + V(t)\, j)] - \Omega(t)\,(y\, i - x\, j)$$

Figure 3.2 Frame of Reference for Unsteady Flow.

direction. We have to make this distinction clear because in our model on convection of core vortices, we break up the calulation into two steps; we first convect the core vortices using the resultant absolute velocity with respect to an inertial coordinate system, followed by determining their positions with coordinates relative to the airfoil-fixed axes.

The unsteady flow Bernoulli's equation for the pressure coefficients on the airfoil surface must be written with respect to the airfoil-fixed coordinate system also. Giesing [Ref. 4] showed this to be written, in our notation, as :

$$C_p = \frac{P - P_\infty}{\frac{1}{2}\rho V_\infty^2} = \left(\frac{V_{stream}}{V_\infty}\right)^2 - \left(\frac{V_{total}}{V_\infty}\right)^2 - \frac{2}{V_\infty^2}\frac{\partial\varphi}{\partial t} \qquad \text{(eqn 3.10)}$$

where V_{stream}, and V_{total} are defined according to Equations 3.7 and 3.9.

Equations 3.8, 3.9, and 3.10 can be correlated to their counter-parts in steady flow, namely 2.6, 2.8 and 2.9 respectively with V_{stream} of Equation 3.7 replacing the V_∞ in Equation 2.8.

C. TIME-DEPENDENT INFLUENCE COEFFICIENTS

1. Definition of Time-Dependent Influence Coefficients

The influence coefficients, A^n_{ij}, A^t_{ij}, B^n_{ij} and B^t_{ij}, involving the source and vorticity distributions described in Section C of Chapter II are still useful. These are indeed time-independent coefficients since they are functions of geometrical parameters which are fixed in our rigid airfoil. Additional influence coefficients involving the shed vorticity panel and the wake core vortices must be defined. These coefficients need to be computed in each time step since their positions vary relative to the airfoil-fixed coordinate system. For that matter, as will be made clear later, those influence coefficients involving the shed vorticity panel must also be computed in every iteration within each time step for the same reasoning.

a. *More A's and B's Influence Coefficients*

Following the notations used previously in steady flow, we define, with the use of the k-subscript to denote time-dependency, additional influence coefficients involving uniformly distributed singularities of source and vorticity. They are the A's and B's coefficients :

- $(B^n_{i,n+1})_k$: normal velocity component induced at the i^{th} panel control point by unit strength vorticity distribution on the shed vorticity panel at time t_k.

- $(B^t_{i,n+1})_k$: tangential velocity component induced at the i^{th} panel control point by unit strength vorticity distribution on the shed vorticity panel at time t_k.
- $(A^x_{n+1,j})_k$: x-velocity component induced at the shed vorticity panel mid point by unit strength source distribution on the j^{th} panel at time t_k.
- $(A^y_{n+1,j})_k$: y-velocity component induced at the shed vorticity panel mid point by unit strength source distribution on the j^{th} panel at time t_k.
- $(B^x_{n+1,j})_k$: x-velocity component induced at the shed vorticity panel mid point by unit strength vorticity distribution on the j^{th} panel at time t_k.
- $(B^y_{n+1,j})_k$: y-velocity component induced at the shed vorticity panel mid point by unit strength vorticity distribution on the j^{th} panel at time t_k.
- $(A^x_{hj})_k$: x-velocity component induced at the center of the h^{th} core vortex by unit strength source distribution on the j^{th} panel at time t_k.
- $(A^y_{hj})_k$: y-velocity component induced at the center of the h^{th} core vortex by unit strength source distribution on the j^{th} panel at time t_k.
- $(B^x_{hj})_k$: x-velocity component induced at the center of the h^{th} core vortex by unit strength vorticity distribution on the j^{th} panel at time t_k.
- $(B^y_{hj})_k$: y-velocity component induced at the center of the h^{th} core vortex by unit strength vorticity distribution on the j^{th} panel at time t_k.
- $(B^x_{h,n+1})_k$: x-velocity component induced at the center of the h^{th} core vortex by unit strength vorticity distribution on the shed vorticity panel at time t_k.
- $(B^y_{h,n+1})_k$: y-velocity component induced at the center of the h^{th} core vortex by unit strength vorticity distribution on the shed vorticity panel at time t_k.

b. New C's Influence Coefficients

The presence of discrete core vortices in the wake requires the definition of new influence coefficients involving point singularity. They are the C's coefficients in our familiar notations :

- $(C^n_{im})_k$: normal velocity component induced at the i^{th} panel control point by unit strength m^{th} core vortex at time t_k.
- $(C^t_{im})_k$: tangential velocity component induced at the i^{th} panel control point by unit strength m^{th} core vortex at time t_k.
- $(C^x_{n+1,m})_k$: x-velocity component induced at the shed vorticity panel mid point by unit strength m^{th} core vortex at time t_k.
- $(C^y_{n+1,m})_k$: y-velocity component induced at the shed vorticity panel mid point by unit strength m^{th} core vortex at time t_k.
- $(C^x_{hm})_k$: x-velocity component induced at the center of the h^{th} core vortex by unit strength m^{th} core vortex at time t_k.

- $(C^y_{hm})_k$: y-velocity component induced at the center of the h^{th} core vortex by unit strength m^{th} core vortex at time t_k.

2. **Computation of Time-Dependent Influence Coefficients**

$(B^n_{i,n+1})_k$ and $(B^t_{i,n+1})_k$ are computed exactly the same way as B^n_{ij} and B^t_{ij} are computed using Equations 2.14 and 2.15 with subscript n+1 replacing j. Similarly, $(A^x_{n+1,j})_k$ and $(A^x_{hj})_k$ are calculated using Equation 2.12 with θ_i set to zero and subsript i appropriately replaced. Also $(A^y_{n+1,j})_k$ and $(A^y_{hj})_k$ are calculated using Equation 2.13 with θ_i set to zero and subsript i appropriately replaced. We do the same for $(B^x_{n+1,j})_k$ and $(B^y_{n+1,j})_k$ using Equations 2.14 and 2.15 respectively and so on for all the rest of A's and B's coefficients. The C's coefficients will be computed with different expressions from those of A's and B's because they are the velocities induced by unit strength core vortex. It can be shown easily, from the geometry of Figure 3.3, that their expressions take on the following forms,

$$(C^n_{im})_k = -\frac{\cos[\theta_i - (\theta_m)_k)]}{2\pi\,(r_{im})_k} \qquad \text{(eqn 3.11)}$$

$$(C^t_{im})_k = -\frac{\sin[\theta_i - (\theta_m)_k)]}{2\pi\,(r_{im})_k} \qquad \text{(eqn 3.12)}$$

where :

$$(r_{im})_k = \surd\,[(xm_i - x_m)^2 + (ym_i - y_m)^2]$$

$$xm_i = \tfrac{1}{2}(x_i + x_{i+1})$$

$$ym_i = \tfrac{1}{2}(y_i + y_{i+1})$$

x_m = x coordinate of m^{th} core vortex at time t_k

y_m = y coordinate of m^{th} core vortex at time t_k

$$\theta_i = \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$$

$$(\theta_m)_k = \arctan\left(\frac{ym_i - y_m}{xm_i - y_m}\right)_k$$

By the same token, $(C^x_{n+1,m})_k$ and $(C^x_{hm})_k$ are computed by Equation 3.11 while $(C^y_{n+1,m})_k$ and $(C^y_{hm})_k$ are computed by Equation 3.12 if θ_i is set equal to zero and the subscript i appropriately replaced.

D. NUMERICAL SOLUTION SCHEME

1. The Flow Tangency Conditions

The flow tangency conditions of Equation 3.5 can be written using the influence coefficients as follows,

$$\sum_{j=1}^{n} [\, A^n_{ij} (q_j)_k \,] + \gamma_k \sum_{j=1}^{n} B^n_{ij} + [\, (V_{stream})_i \cdot n_i \,]_k$$

$$+ (\gamma_w)_k (B^n_{i,n+1})_k + \sum_{m=1}^{k-1} [\, (C^n_{im}) (\Gamma_{m-1} - \Gamma_m) \,] = 0 \;, \quad i = 1,2,\ldots,n \qquad \text{(eqn 3.13)}$$

where $(V_{stream})_i$ is evaluated by Equation 3.7 at the i^{th} panel control point

This equation, though it seems complex, says nothing more than summing to zero all the velocity contributions due to individual singularity. Substituting $(\gamma_w)_k$ from Equation 3.2, collecting like terms and rearranging the equation into,

$$\sum_{j=1}^{n} [\, A^n_{ij} (q_j)_k \,] = \gamma_k [\, (\ell/\Delta_k) (B^n_{i,n+1})_k - \sum_{j=1}^{n} B^n_{ij} \,]$$

$$\{ - [\, (V_{stream})_i]_k \cdot n_i \,] - (\ell/\Delta_k)\, \gamma_{k-1} (B^n_{i,n+1})_k$$

$$- \sum_{m=1}^{k-1} [\, (C^n_{im})_k (\Gamma_{m-1} - \Gamma_m) \,] \} \;, \quad i = 1,2,\ldots,n \qquad \text{(eqn 3.14)}$$

2. The Iterative Solution Procedure

Equation 3.14 is arranged in this form because we intend to solve $(q_j)_k$ ($j = 1,2,\ldots,n$) in terms of γ_k. Expanding Equation 3.14 for $i = 1,2,\ldots,n$ results in the following matrix equation,

$$[\, A \,] \{\, q \,\}_k = \gamma_k \{\, B \,\}_k + \{\, C \,\}_k \qquad \text{(eqn 3.15)}$$

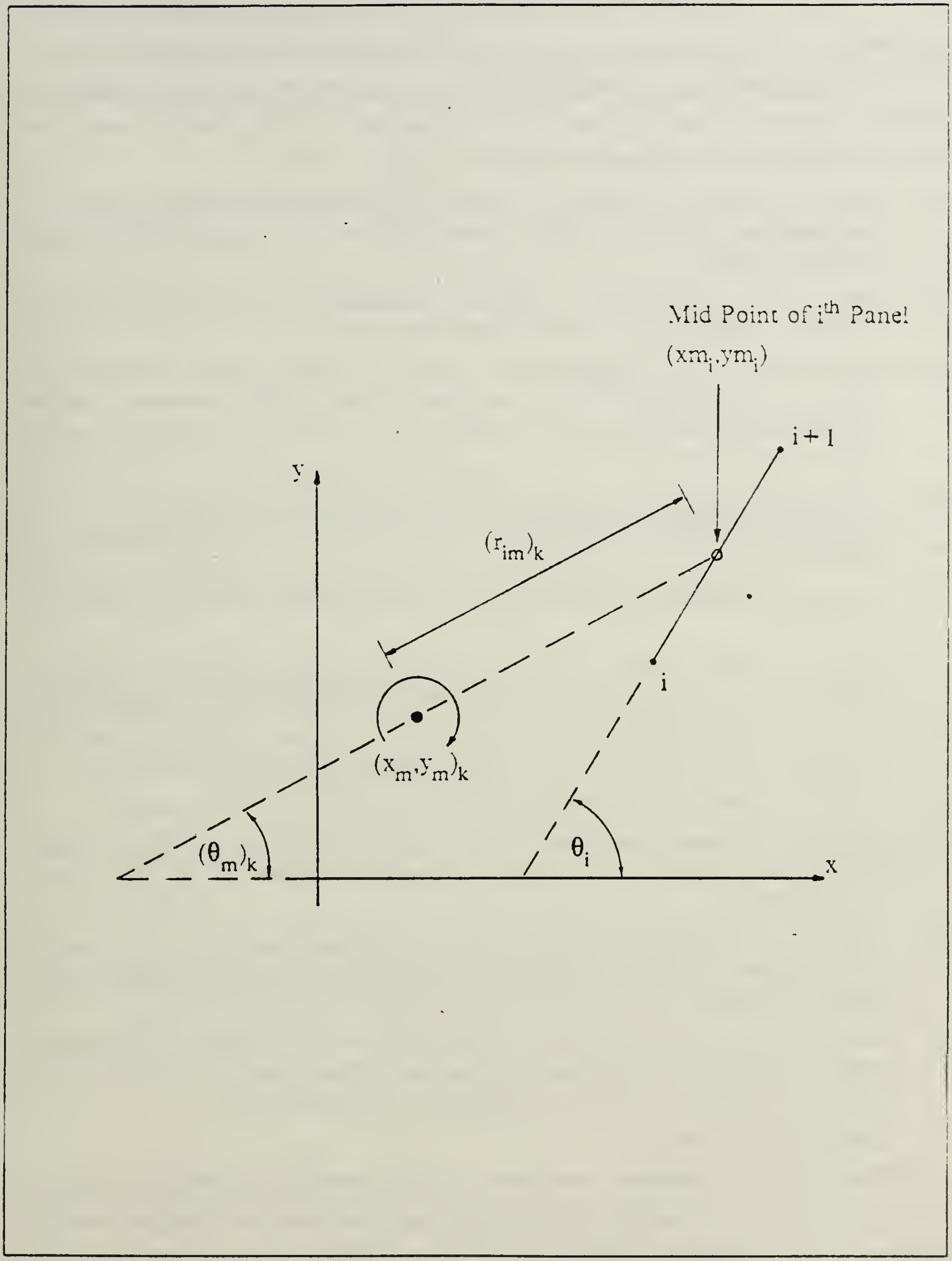


Figure 3.3 Influence Coefficients due to Point Singularities.

where [A] is an $n \times n$ matrix whose elements are known constants. $\{ B \}_k$ and $\{ C \}_k$ are $n \times 1$ column vectors whose elements are known *only* if the shed vorticity panel at t_k is established; that is, if Δ_k and Θ_k are known, then we can calculate all the influence coefficients on the right-hand-side of Equation 3.14. We therefore make use of this idea to formulate our iterative solution procedure as follows :

(1) Project the wake core vortices downstream according to the time step and the local resultant velocities at their respective centers with respect to an inertial coordinate system.

(2) Compute the coordinates of these core vortices relative to the airfoil-fixed coordinate system due to its time-dependent motion.

(3) Start iteration cycle for current time step by initially assuming some guess values of Δ_k and Θ_k. We can use, except for the first time-step, values obtained at previous time step. Compute then the influence coefficients needed in Equation 3.14 or 3.15.

(4) Obtain $(q_j)_k$ in terms of γ_k by solving Equation 3.15 as a linear system with two right-hand-sides by the same Gaussian elimination algorithm used in steady flow.

(5) Calculate the tangential velocities at the trailing edge panels, all in terms of γ_k.

(6) Invoke the Kutta condition of Equation 3.6 (with some efforts in algebriac manipulation) to solve for γ_k since it is the only unknown in that equation.

(7) Once γ_k is solved, $(q_j)_k$ are then known. We can then calculate the velocity components $(U_w)_k$ and $(V_w)_k$ at the mid point of the shed vorticity panel.

(8) Equation 3.3 and 3.4 hence enable us to update the values of Δ_k and Θ_k.

(9) Repeat the iteration cycle from steps (3) to (9) until converged values of Δ_k and Θ_k are obtained. Alternatively convergence can be set for $(U_w)_k$ and $(V_w)_k$ instead.

(10) Compute the tangential velocities and disturbance potential at all panel control points in order to determine the pressure distribution which can be integrated to give forces and moments.

(11) Compute the resultant velocities which occur at the centers of all the core vortices that will be convected down-stream. These velocities must be the absolute velocities with respect to an inertial coordinate system.

3. Computation of Velocities

The iterative procedure mentioned in the previous subsection requires calculation of tangential velocities at the trailing edge panels and the absolute velocity components $(U_w)_k$ and $(V_w)_k$. They are computed differently due to the use of a different frame of reference.

a. Tangential Velocities on Airfoil Panels

The tangential velocities $[(V^t)_i]_k$ (i= 1,2,....,n) at all the panel control points are calculated using the airfoil-fixed coordinate frame of reference as follows :

$$[(V^t)_i]_k = \sum_{j=1}^{n} [A^t_{ij} (q_j)_k] + \gamma_k \sum_{j=1}^{n} B^t_{ij}$$

$$+ [(V_{stream})_i \cdot t_i]_k + (\gamma_w)_k (B^t_{i,n+1})_k$$

$$+ \sum_{m=1}^{k-1} [(C^t_{im})_k (\Gamma_{m-1} - \Gamma_m)] \ , \qquad i=1,2,....,n \qquad \text{(eqn 3.16)}$$

b. Core Vortices Convection Velocities

The resultant velocities at all core vortices are calculated using the inertial frame of reference fixed with respect to V_∞ but resolving them into components in the directions coincide with the airfoil-fixed coordinate system as shown below :

$$(U_h)_k = \sum_{j=1}^{n} [(A^x_{hj})_k (q_j)_k] + \gamma_k \sum_{j=1}^{n} (B^x_{hj})_k$$

$$+ (V_\infty \cdot i)_k + (\gamma_w)_k (B^x_{h,n+1})_k$$

$$+ \sum_{\substack{m=1 \\ m \neq h}}^{k-1} [(C^x_{hm})_k (\Gamma_{m-1} - \Gamma_m)] \qquad \text{(eqn 3.17)}$$

$$(V_h)_k = \sum_{j=1}^{n} [(A^y_{hj})_k (q_j)_k] + \gamma_k \sum_{j=1}^{n} (B^y_{hj})_k$$

$$+ (V_\infty \cdot j)_k + (\gamma_w)_k (B^y_{h,n+1})_k$$

$$+ \sum_{\substack{m=1 \\ m \neq h}}^{k-1} [(C^y_{hm})_k (\Gamma_{m-1} - \Gamma_m)] \qquad \text{(eqn 3.18)}$$

Notice the use of V_∞ instead of V_{stream} in Equations 3.17 and 3.18. Also the subscript h is just an index usable for any core vortex. We can obtain $(U_w)_k$ and $(V_w)_k$ if h is replaced by $n+1$ in these equations.

4. **Disturbance Potential and Pressure Distribution**

a. *Why We Need the Disturbance Potential*

The concept of disturbance potential φ has been instrumental in the formulation of both the steady and unsteady flow problems. However, it has never gone beyond using it merely as a vehicle to understanding the superposition of simple flows. The disturbance potential need not be solved for at all in the steady flow problem formulation. This is because what one really is going after is the spatial derivative of this disturbance potential, i.e. the disturbance induced velocity, from which the pressure distribution can be obtained. We have, in all our solutions so far, been successful in avoiding any disturbance potential calculation since the concept of influence coefficients allows us a direct evaluation of the velocity. Unfortunately, as can be seen in Equation 3.10, when we proceed further to compute the pressure distribution on the airfoil surface in unsteady flow, we are faced with the problem of evaluating the disturbance potential φ, or more precisely the rate of change of φ, which we approximate by using a backward finite difference expression. Therefore, the pressure coefficients at the i^{th} panel control point can be rewritten, in terms of non-dimensional variables, as,

$$[(C_p)_i]_k = [(V_{stream})_i]_k^2 - [(V^t)_i]_k^2 - 2\,\frac{(\varphi_i)_k - (\varphi_i)_{k-1}}{t_k - t_{k-1}} \qquad \text{(eqn 3.19)}$$

where $(V^t)_i$ is calculated by Equation 3.16 and $(V_{stream})_i$ is the non-dimensional (by V_∞) form of Equation 3.7 evaluated at the i^{th} panel control point.

We thus need to calculate at each time step, the disturbance potential at all the panel control points. Short of having to solve the Laplace's equation by a finite difference scheme, we evaluate the disturbance potential φ by integrating the velocity field in two stages from upstream at infinity to the airfoil leading edge, then along the airfoil surface from the leading edge to each panel control point. Care must be taken here to include only the velocity contribution due to disturbances.

One important question arises, in this approach, as to what value of disturbance potential we should use at infinity before we carry out the line integral. We

must therefore analyse the behaviour of φ at infinity by examining the singularities that constitute the disturbance. They are the source and vorticity distributions on the airfoil surface and the core vortices in the wake. These singularities induce no velocity at infinity from the knowledge of simple flows. In other words, the disturbance potential φ at infinity is independent of spatial coordinates. The next question we should ask is whether φ at infinity is time-dependent? Let us adopt the view-point that if we are at infinity looking at our airfoil and its associated wake, we simply see a point vortex with a total circulation Γ_0 at time t_0. We have already identified that Γ_0 remains constant by Helmholtz's theorem. It only gets redistributed, as time progresses, over the airfoil surface and in the wake. Notice that the previous statement regarding what one would see at infinity said nothing about the source distributions. The source distributions though vary (or get redistributed) as the time progresses, the total source strength necessarily remains zero at all time in order to enforce a closed contour representing the airfoil thickness. This is also the reason why the unsteady flow solution needs an additional model to handle the vorticity conservation since the source conservation is already implicitly so for a closed contour to exist. From these discussions, we are certain that the disturbance potential φ at infinity is an absolute constant (independent of time and spatial coordinates) whose value is fixed only by the initial condition we decide to start solving the unsteady problem. The actual value of φ at infinity is in fact immaterial so long as we know it is constant because its value disappears conveniently as we subtract $(\varphi_i)_{k-1}$ from $(\varphi_i)_k$ in Equation 3.19.

b. Computation of Disturbance Potential

We begin by choosing an arbitrary straight line extending upstream to infinity from the leading edge of the airfoil along a direction parallel to V_∞. For practical purposes, we set infinity at say ten chord lengths away from the leading edge since the velocities induced, at field points thereafter, by the disturbances are small enough to be negligible. This line is divided into z panels with element lengths near the leading edge comparable to the panel sizes used on the airfoil. However, the panel size is progressively increased to take advantage of the inversely decaying induced velocities at larger distances. We compute the tangential components of the induced velocities at the mid points of these panels using influence coefficients analogous to those used on the airfoil panels. Using subscript f to denote these panel mid-points, we can define influence coefficients $(A^t_{fj})_k$, $(A^t_{f,n+1})_k$, $(B^t_{fj})_k$, $(B^t_{f,n+1})_k$, and $(C^t_{fm})_k$ and compute them using the same expressions for calculating the A's, B's and C's coefficients used

before with $\cos\theta_i$ replaced by $(-\cos\alpha)$, $\sin\theta_i$ replaced by $(-\sin\alpha)$ and subscript i replaced by f. With the help of these influence coefficients, the tangential velocity induced by disturbances at the f^{th} panel mid point is :

$$[(V^t_{\varphi})_f]_k = \sum_{j=1}^{n} [(A^t_{fj})_k (q_j)_k] + \gamma_k \sum_{j=1}^{n} (B^t_{fj})_k$$

$$+ (\gamma_w)_k (B^t_{f,n+1})_k + \sum_{m=1}^{k-1} [(C^t_{fm})_k (\Gamma_{m-1} - \Gamma_m)] \qquad \text{(eqn 3.20)}$$

valid for f= 1,2,....,z. The disturbance potential at the airfoil leading edge is the sum of the products of the disturbance induced velocity at each panel and the panel length.

$$(\varphi_{le})_k = - \sum_{f=1}^{z} [(V^t_{\varphi})_f]_k \sqrt{[(x_{f+1} - x_f)^2 + (y_{f+1} - y_f)^2]} \qquad \text{(eqn 3.21)}$$

Similarly, for the line integral over the airfoil surface, we compute the tangential component of the disturbance induced velocity at the i^{th} panel control point using the following equation :

$$[(V^t_{\varphi})_i]_k = \sum_{j=1}^{n} [A^t_{ij} (q_j)_k] + \gamma_k \sum_{j=1}^{n} B^t_{ij}$$

$$+ (\gamma_w)_k (B^t_{i,n+1})_k + \sum_{m=1}^{k-1} [(C^t_{im})_k (\Gamma_{m-1} - \Gamma_m)] \qquad \text{(eqn 3.22)}$$

which is valid for i= 1,2,....,n. Performing the line integration by summation, the disturbance potential at the i^{th} nodal point on the airfoil is :

$$(\varphi_{node\ i})_k = (\varphi_{le})_k + \sum_{j=i_{le}}^{i-1} [(V^t_{\varphi})_j]_k \, r_{j,j+1} , \qquad \text{for } n > i \geq i_{le}$$

$$= (\varphi_{le})_k - \sum_{j=i}^{i_{le}-1} [(V^t_{\varphi})_j]_k \, r_{j,j+1} , \qquad \text{for } i_{le} > i \geq 1 \qquad \text{(eqn 3.23)}$$

where $r_{j,j+1}$ denotes the panel length.

$$r_{j,j+1} = \sqrt{[(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2]}$$

Finally, the disturbance potential at the i^{th} panel control point is,

$$(\varphi_i)_k = \tfrac{1}{2} [(\varphi_{node\ i})_k + (\varphi_{node\ i+1})_k] , \qquad i = 1,2,....,n \qquad \text{(eqn 3.24)}$$

5. **Computation of Forces and Moments**

The C_ℓ, C_d and C_m about the leading edge are calculated in exactly the same way as it is done for the steady flow problem by integrating the pressure distribution (See section D-4 of Chapter II).

E. FLOW MODELING OF SHARP EDGE GUST FIELD

The unsteady flow solution described so far can be extended to the study of airfoils penetrating a sharp edge gust by modifying the boundary conditions with the assumption that the gust front remains straight while passing through the airfoil. The same assumption has been used in both [Ref. 3] and [Ref. 4]. An additional model in [Ref. 3] using distribution of singularities along the gust front had successfully attempted to simulate the distortion of the gust front passing over the airfoil surface. It was shown that the pressure distributions, during the time when the gust front remained on the airfoil surface, were affected only at the neighbourhood of the gust front. The overall pressure upstream and downstream of the gust front stayed essentially the same. The distorted gust front model is not used in program U2DIIF. The use of the relatively simple yet sufficiently accurate model of a straight gust front affords the modifications to the unsteady flow solution to be confined only to the flow tangency conditions. That is to say, the expression of V_{stream} in Equation 3.7 would include the gust velocity for panels that are already in the gust field during the penetration phase. Similarly, the computation of core vortex velocities using Equations 3.17 and 3.18 have the gust velocity added to the V_∞ if the core vortices are already in the gust field.

In an attempt to generalise the solution for cases where airfoils enter the gust field at an angle of attack, the convenient model used in [Ref. 3] by setting the computation to proceed, for the undistorted gust front simulation, so that the gust front always coincides with the nodal points is difficult to implement. At any one time,

if an airfoil enters a gust field at an angle of attack, the gust front would appear in between two nodes of a particular panel on one surface while the gust front proceeds from node to node on the other surface. We therefore further modify the flow tangency condition only on that particular panel where the gust front lies in between two nodes by taking the gust velocity on that panel to be proportional to the fraction of panel length partially submerged in the gust field.

IV. DESCRIPTION OF COMPUTER CODE U2DIIF

A. PROGRAM U2DIIF STRUCTURES AND CAPABILITIES

1. Restrictions and Limitations

The numerical formulations of both the steady and unsteady flow problems outlined in the previous Chapters are coded in a FORTRAN computer program called U2DIIF (See Appendix A for the program listings). The present solution methods treat the inviscid and incompressible flow as an approximation to the real flow so long as the viscous effect is negligible and the flow stays attached on the airfoil surface at all time. These restrictions are no strangers to any one who is familiar with any other potential flow solution methods. Other than the implicit restrictions of potential flow solution, the method is entirely general in that the shape of airfoil is arbitrary and any arbitrary continuous motion of the airfoil could be simulated using either the closed form (i.e. explicit equations) or discrete data points to describe the time-history of the translational and rotational velocities.

The storage of the computer that carries out the calculations may be the other limitation one should consider. The storage requirements grow rapidly with the number of panels (n) and the number of computation time steps (m). By far the prime contributor to this storage requirement comes from the massive amount of influence coefficients. The number of influence coefficients increases with the square of the number of panels (n^2). Each time step increment adds ($2n + m^2$) more influence coefficients due to the formation of shed vorticity. The current program fixes the maximum number of airfoil panels to 200 and the maximum allowable time steps is also 200.

An additional constraint worth mentioning concerns the gust field simulation whereby the current solution methodology is valid except in the use of the same pressure equation arising from the unsteady Bernoulli's equation (See Equation 3.10). The fundamental assumption underlying the derivation of this equation is the irrotationality of the flow field. There is no doubt that the flow fields upstream and downstream of the gust front are irrotational. However, when one needs to obtain the pressure on the airfoil surface, an implicit integration is done across the gust front. A flow field inclusive of the gust front is rotational since the line integral of velocity in a

closed path does not vanish when the gust front is crossed. Failing the proper derivation of a new pressure equation applicable to unsteady rotational flows, care must be exercised to regard the present method as an approximate solution to gust fields of weak strengths only.

2. Current Structures of U2DIIF MAIN Program

The overall program logic-flow chart is as shown in Figure 4.1. The program first reads in the input data from filecode 1 and sets up the airfoil panel nodes and slopes. Immediately after that, the steady flow calculations are executed for the initial angle of attack α_i according to the solution scheme described in Section D of Chapter II. The steady flow solution is included primarily to :

- Provide the necessary initial parameters for the unsteady flow solution to proceed in time. In other words, the steady flow solution handles the V_∞ and initial angle of attack α_i one decides to begin the unsteady flow calculation.
- Allow the code to function directly as a steady flow solver as and when necessary without having to do the time consuming unsteady flow iterative solution and approach the steady flow as time approaches infinity.

The program terminates once the steady flow calculations are done if the program determines, based on the input data set by user, no requirement for unsteady flow solutions. Otherwise the unsteady flow calculations will be activated by selecting and computing the rigid body motions of the airfoil and the corresponding computation time-step size. Currently, all the time dependent motions are equation-generated, they are the positions and rates of the translational and rotational motions. Incorporated as case-runs within the program U2DIIF are the following motions :

(1) Step change in angle of attack from any initial value.

(2) Modified-ramp change in angle of attack about any pivot point from any initial value.

(3) Harmonic translational motion at any angle of attack.

(4) Harmonic rotational motion about any pivot point at any mean angle of attack.

(5) Sharp edge gust penetration at any angle of attack.

Should one decide to generate the airfoil's motion using discrete data points as a function of time, the program could be easily modified.

The computation time-step sizes for the harmonic translational and rotational motions are constant values determined by the frequencies (FREQ) and the number of computation per cycle (DTS). For the case of step change in angle of attack, the computation time-step size is progressively increasing, from a starting value (DTS), as

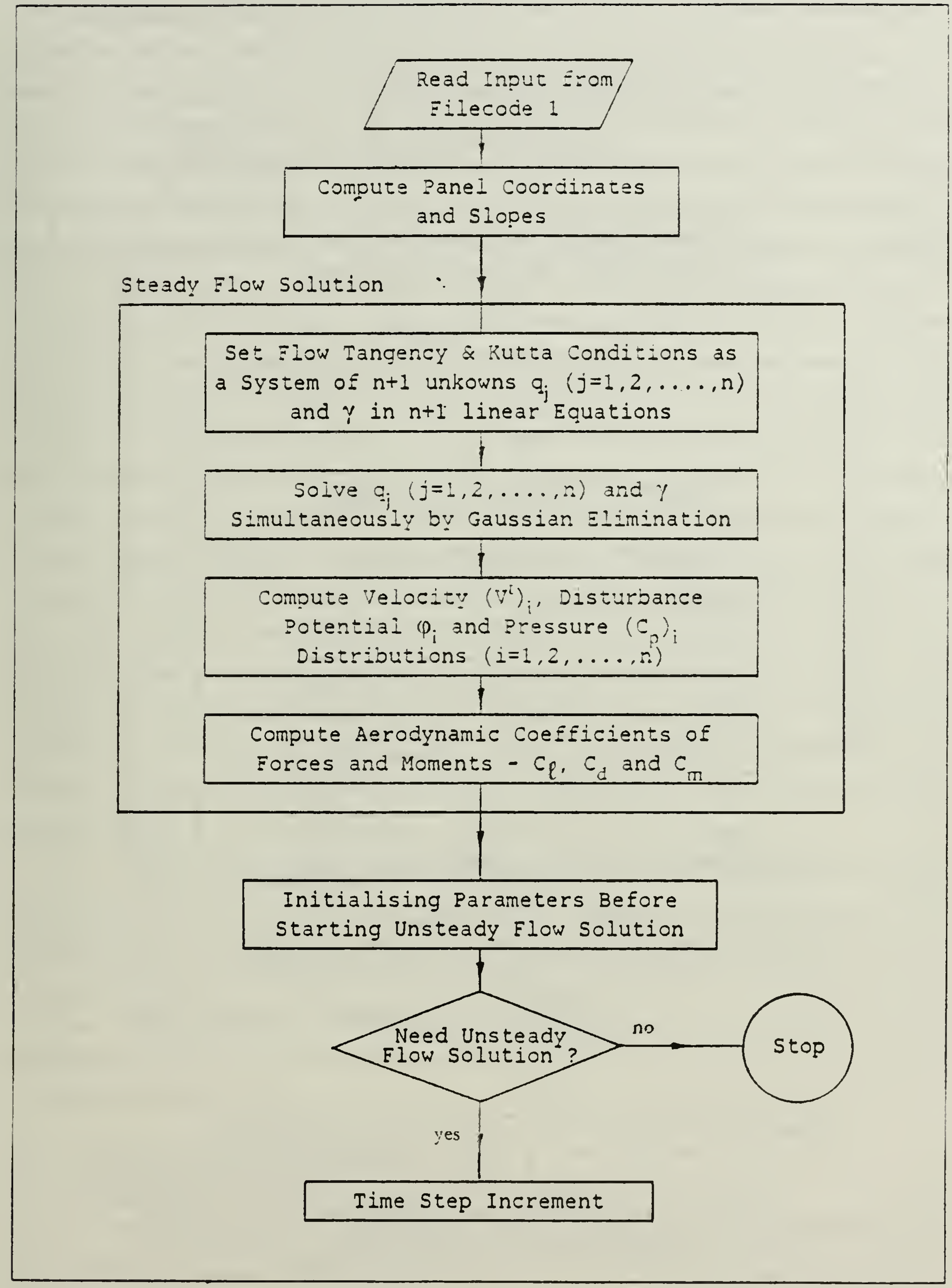


Figure 4.1 Flow Chart for U2DIIF Computer Code.

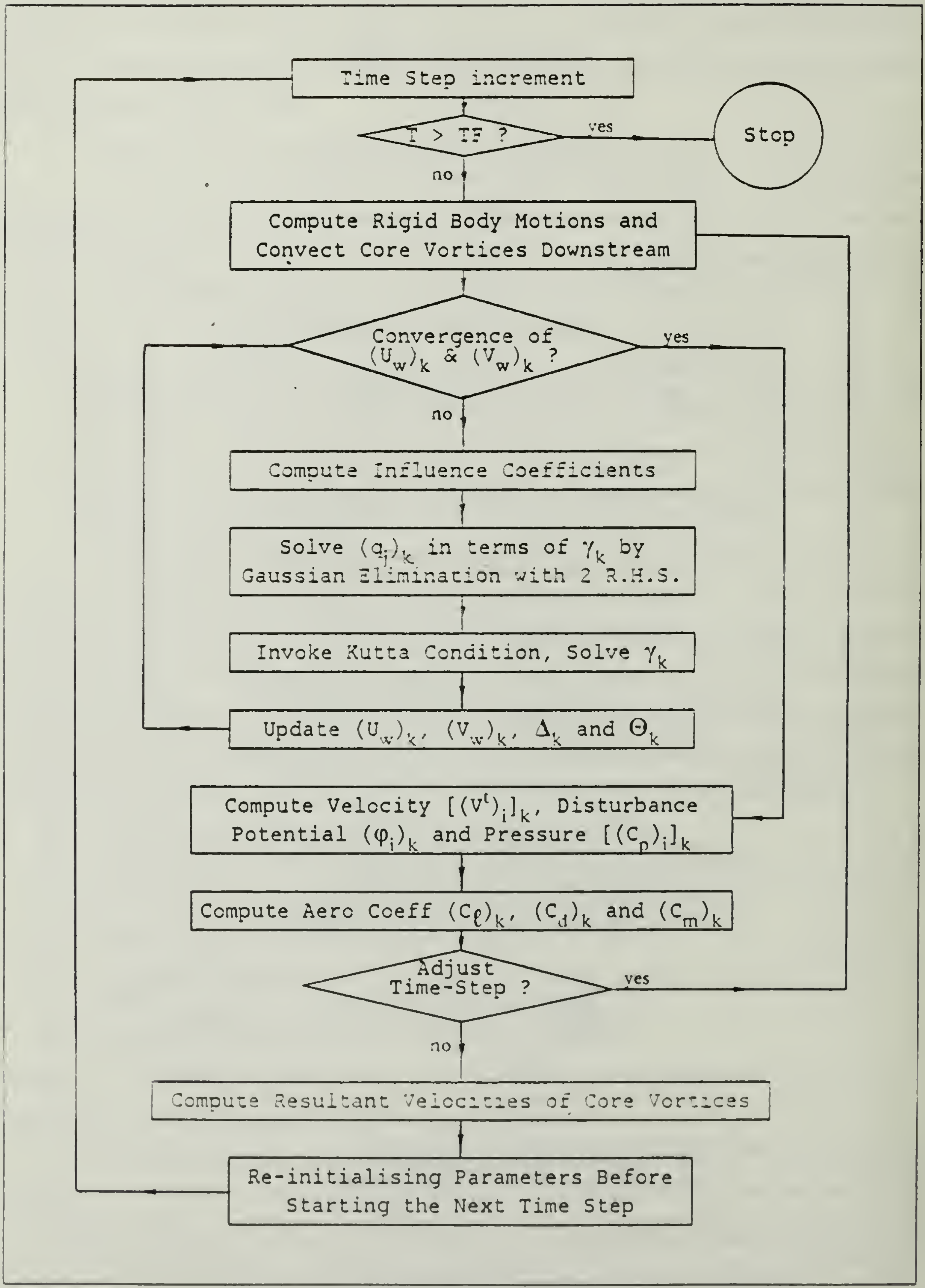


Figure 4.1 . (cont'd.)

time increases. The case of the modified ramp change in angle of attack adopts initially a constant computation time-step size (DTS) during the transient rising of the angle of attack. Once the final angle of attack is reached, the computation time-step size is progressively increased also. Similarly for the case of airfoil penetrating a gust field, the computation time-step size (DTS) is constant during the period when the gust front remains on the airfoil surface but progressively increases once the entire airfoil is submerged in the gust field. These variations in computation time steps are to provide greater flexibility both in capturing transients and covering relatively large total time of computation without having to contend with the storage space requirements described previously. These variations in time-step sizes described so far are associated with setting the input parameter TADJ to zero. If TADJ is chosen to be non-zero, all the case-runs would compute initially using the starting time-step sizes, based on DTS for non-oscillatory motions and FREQ & DTS for harmonic motions, and the program would prompt for an user choice of time step adjustment. If the answer is yes, the program would back-track the previous solution and recompute the current solution using an adjusted time-step size that is TADJ times the initial value (DTS). This special time step variation feature gives the program added capability of allowing an interactive time step selection during the progress of unsteady flow computation. The ability to back-track and recompute the current solution using a different time-step size enhances the possibility of using program U2DIIF together with a viscous flow solver forming an Inviscid-Viscous-Interactive solution scheme which often requires such time step variations.

The MAIN program performs the iterative solution procedures set out in Section D of Chapter III. The convergence check during the iterative solution is done through the user specified tolerance between successive iterative solutions of both $(U_w)_k$ and $(V_w)_k$. The solution continues into the next time-step by selecting the time step size according to the particular case-run and projecting all the wake core vortices downstream so that their new positions relative to the airfoil at the new time step can be correctly determined.

B. DESCRIPTION OF SUBROUTINES

1. Subroutine BODY

This subroutine is called by subroutine SETUP if the user selects an airfoil that is either a NACA XXXX or 230XX type. It in turn calls subroutine NACA45 to obtain the airfoil thickness and camber distributions and returns with the computed (x,y) coordinates of the panel nodal points.

2. **Subroutine COEF**

This subroutine is called by the MAIN program in the unsteady flow calculations. It utilises, at each iteration cycle, the influence coefficients generated by subroutine INFL to calculate the coefficients of the matrix Equation 3.15 by expanding Equation 3.14. These matrix coefficients are necessarily set up in this way so that the source strengths could be solved in terms of the vorticity strength by subroutine GAUSS as a linear system with two right-hand-sides.

3. **Subroutine COFISH**

This subroutine is called by the MAIN program to set up the coefficients of the matrix system of Equation 2.18 for steady flow where the source strengths and vorticity strength are solved simultaneously by subroutine GAUSS as a linear system with one right-hand-side. The matrix coefficients are calculated using Equations 2.16 and 2.17.

4. **Subroutine CORVOR**

This subroutine is called by the MAIN program at nearing the end of the unsteady flow calculations before starting a new time step. It computes the resultant convective velocities for all the wake core vortices with respect to an inertial frame of reference according to Equations 3.17 and 3.18 where all the appropriate influence coefficients are linked through common block from subroutine INFL.

5. **Subroutine FANDM**

This subroutine is used in both the steady and unsteady flow calculations. It is called by the MAIN program immediately after the pressure distribution over the airfoil panels are known so that it can perform the simple integration of pressure in the appropriate directions to give the aerodynamic force and moment coefficients of lift, drag and pitching moment about the leading edge according to Equations 2.22 through 2.26.

6. **Subroutine GAUSS**

This subroutine is the standard linear system solver that employs the well known Gaussian elimination with partial pivoting and operates simultaneously on a user specified number of right-hand-sides. It is called by the MAIN program in both the steady and unsteady flow calculations. In order to use GAUSS, the coefficients of the augmented matrix must be set up so that GAUSS will return the solutions replacing the corresponding columns of the augmented matrix that were initially occupied by the right-hand-sides. The coefficient set-ups are done by subroutines COFISH and COEFF respectively for the steady and unsteady flow problems.

7. **Subroutine INDATA**

This subroutine is called by the MAIN program to read in the first three sets of data cards and returns to the MAIN program if IFLAG $\neq 0$. Otherwise it continues to read in the fourth data card as the NACA number corresponding to the type of airfoil and calculates the thickness parameters that will be used by subroutine NACA45.

8. **Subroutine INFL**

This subroutine is the generator for all the influence coefficients that need to be stored and used by many subroutines associated with the unsteady flow calculations. It utilises the known relative geometrical parameters of the singularities to carry out computations based on Equations 2.12 through 2.15, 3.11 and 3.12. The MAIN program calls this subroutine in every iteration cycle of each time step so that the time-dependent coefficients can be updated as and when necessary. Time-independent coefficients are computed only once in the entire unsteady flow solutions. Those influence coefficients involving the wake core vortices are updated in each time step while those involving the shed vorticity panel are calculated as frequently as the number of iterations take to terminate a converged solution. It, however, does not compute and store those influence coefficients needed for the determination of disturbance potential (Equation 3.20) simply because they are used only once in each time step.

9. **Subroutine KUTTA**

This subroutine is called, in the unsteady flow calculations, by the MAIN program during every iteration cycle in each time step to invoke the Kutta condition for unsteady flow expressed in Equation 3.6. It calculates the tangential velocities at the trailing edge panels using Equation 3.16 in terms of the unknown vorticity strength that is manipulated and solved algebraicly.

10. **Subroutine NACA45**

This subroutine is called by subroutine BODY if the airfoil selected belongs to the families of NACA 4-digits airfoils or the NACA 5-digits airfoils of type 230XX who share common thickness distributions with the 4-digits airfoils having the same thickness to chord ratio. The thickness and camber distribution data of these airfoils are calculated and returned to BODY.

11. **Subroutine PRESS**

This subroutine is called by the MAIN program to calculate the pressure distribution over the airfoil panels after the iterative solution for the unsteady flow problem has successfully met the convergence criterion. It first computes the tangential velocities at all panel control points using Equation 3.16, then performs the disturbance potential evaluation at the current time step according to Equations 3.20 through 3.24. Together with the disturbance potential data obtained from the previous time step, it calculates the pressure distribution using Equation 3.19.

12. **Subroutine SETUP**

This subroutine sets up the panel nodal coordinates for MAIN program by reading the 4^{th} and 5^{th} data sets of the input file if IFLAG = 1 is set. It skips the data reading if IFLAG = 0 and proceeds to set up the node distribution and call subroutine BODY to calculate the airfoil coordinates. The node distribution adopts a cosine formula in order to have closely packed panels towards the leading and trailing edges for improvements in solution accuracy. Regardless of how the nodal coordinates are obtained, SETUP determines the panel slopes and airfoil perimeter length.

13. **Subroutine TEWAK**

This subroutine is called by the MAIN program at every iteration cycle of each time step of the unsteady flow calculations to compute the resultant velocity components at the mid point of the shed vorticity panel using Equation 3.17 and 3.18 These velocity components are necessary to ensure the correct establishment of the shed vorticity panel length and orientation which governs the successful implementation of the iterative solution scheme for the unsteady flow problems.

14. **Subroutine VELDIS**

This subroutine returns to the MAIN program the velocities and pressure distributions for steady flow calculation using Equations 2.20 and 2.21. It also performs the evaluation of the disturbance potential at the panel control points. Though these disturbance potential data are not necessary for steady flow solution, they will be needed in the first time step of the unsteady flow pressure calculation.

C. INPUT DATA FOR PROGRAM U2DIIF

Program U2DIIF reads its input data from filecode 1. An example of the input data file is as shown in Appendix B for the case where the airfoil nodal coordinates are input by user. User could however let the program generate the nodal coordinates if

the airfoil chosen happens to belong to the family of NACA 4-digits or 5-digits of type 230XX. To do this, simply change IFLAG to zero in the first item of the 3^{rd} set of data card and replace the nodal coordinates data in the 4^{th} and 5^{th} sets of data cards by a single data card containing only the particular airfoil's NACA number using Format (I5). Figure 4.2 contains an itemised description of the sequential input variables.

D. OUTPUT DATA FROM PROGRAM U2DIIF

Appendix C contains a sample output data generated by using the input data set shown in Appendix B. Due to the repetitive nature of output as the computation time progresses, only data at selective time steps are shown. The output data file begins with writing out what the program has read from the input data file followed by the computed nodal coordinates only if they are program generated, otherwise proceeds to write the computed airfoil perimeter length. The next set of output data are the steady flow solution parameters of distributed source strengths, vorticity strength, pressure and velocity distributions as well as the force and moment coefficients. The output data terminates at this point unless unsteady flow solution is required. It then prints, for each time step, the unsteady flow solution parameters similar to the previous output for steady flow with additional information pertaining to the rigid body motions and trailing wake vortices data. An explanation of the output variable names are listed in Figure 4.3. All output parameters are non-dimensional quantities.

Data Set #1	Format (I5) - 1 data card
ITITLE	- Number of title cards to be used in Data Set #2.
Data Set #2	Format (20A4) - ITITLE data cards
TITLE	- Headings to be printed on output for case run identification.
Data Set #3	Format (3I5) - 1 data card
IFLAG	- 0 if airfoil is NACA XXXX or 230XX. - 1 otherwise.
NLOWER	- Number of panels used on airfoil lower surface.
NUPPER	- Number of panels used on airfoil upper surface (need not be the same as NLOWER).
Data Set #4	Format (6F10.6) if IFLAG = 1 - variable data cards
X(I)	- x-nodal coordinates (divided by the chord length, c). A total of n + 1 nodal points divided into 6 points per data card.
Data Set #5	Format (6F10.6) - variable data cards.
Y(I)	- y-nodal coordinates (divided by c) corresponding to the Data Set #4 if IFLAG = 1.
Data Set #6	Format(7F10.6) - 1 data card
ALPI	- Initial angle of attack (AOA) in deg.
DALP	- Increment in AOA in deg for non-oscillatory motions. - Maximum amplitude of AOA change in deg for rotational harmonic motions.
TCON	- Non-dimensional rise time ($V_\infty t/c$) of AOA for motion involving modified-ramp change in AOA.
FREQ	- Non-dimensional oscillation frequency ($\omega c/V_\infty$) for harmonic motions.
PIVOT	- The length from the pivot point to the leading edge divided by c (postive aft) for rotational motions.
UGUST	- Magnitude of gust velocity (divided by V_∞) along V_∞.
VGUST	- Magnitude of gust velocity (divided by V_∞) perpendicular to V_∞
Data Set #7	Format (3F10.3) - 1 data card
DELHX	- Amplitude of chordwise translational oscillation divided by c. (positive forward).
DELHY	- Amplitude of transverse translational oscillation divided by c. (positive downward).
PHASE	- Phase angle in deg between the chordwise and transverse translational oscillation with the latter as reference.
Data Set #8	Format (4F10.3) - 1 data card
TF	- Final non-dimensional time to terminate unsteady flow solution.
DTS	- Starting time-step size for non-oscillatory motions if TADJ = 0.0. - Number of computation steps per cycle for harmonic motions. - Baseline time-step size for all motions if TADJ ≠ 0.0.
TOL	- Tolerance criterion for checking the convergence between successive iterations of $(U_w)_k$ and $(V_w)_k$
TADJ	- Factor by which DTS will be adjusted.

Figure 4.2 List of Input Variables.

TK	- Time step t_k.
TKM1	- Time step t_{k-1}.
ALPHA(T)	- Angle of attack at time t_k.
OMEGA(T)	- Rotational velocity (positive counter clockwise) at time t_k.
U(T)	- Chordwise translation velocity (postive forward) at time t_k.
V(T)	- Transverse translational velocity (positive downward) at time t_k.
NITR	- Iteration number.
VXW	- Iterative solution of $(U_w)_k$.
VYW	- Iterative solution of $(V_w)_k$.
Wake	- Iterative solution of shed vorticity panel length Δ_k.
THETA.	- Iterative solution of shed vorticity panel orientation Θ_k.
GAMMA	- Iterative solution of the strength of vorticity distribution.
J	- Panel number.
X(J)	- x-coordinate of the mid point of j^{th} panel.
Q(J)	- Strength of source distribution on the j^{th} panel.
CP(J)	- Pressure coefficient at the mid point of j^{th} panel.
V(J)	- Total tangential velocity at the mid point of j^{th} panel referenced to the airfoil-fixed coordinate system.
CD	- Drag coefficient.
CL	- Lift coefficient.
CM	- Pitching moment coefficient about the leading edge.
M	- Trailing wake core vortex number.
X(M)	- x-coordinate of the center of m^{th} core vortex.
Y(M)	- y-coordinate of the center of m^{th} core vortex.
CIRC	- Circulation strength of the m^{th} core vortex.

Figure 4.3 List of Output Variables.

V. RESULTS AND DISCUSSIONS ON CASE-RUNS

This Chapter presents the results of numerous case-runs of U2DIIF code for the purpose of verifying the code. The various airfoils used in the case-runs are deliberately chosen to be the same as those airfoils where direct comparison of results can be made with either theoretical analyses and/or experimental data available in the literature.

A. STEP CHANGE IN ANGLE-OF-ATTACK

1. Case-Run Definitions

Consider an airfoil initially at zero angle of attack to the free stream V_{∞} that undergoes a step change in angle of attack at t_0. The resulting flow should then provide the time-dependent information on the build-up of aerodynamic forces and moments on the airfoil resembling the classical results of Wagner [Ref. 5] calculated based on linearised theory. Although Wagner prescribed a slightly different initial condition in that the airfoil is initially at rest and impulsively started at an angle of attack and velocity V_{∞}, the difference is insignificant, especially for a symmetrical airfoil. This is because the seemingly different initial conditions when translated into the mathematical model means that the step change in AOA uses non zero initial circulation Γ_0 at t_0 with non zero initial disturbance potential at infinity if the airfoil is cambered. For a symmetric airfoil, these initial values are all zeroes and therefore mathematically would be the same as the initial conditions prescribed by Wagner.

2. Results and Discussions

a. Von Mises 8.4% Thick Symmetrical Airfoil

A 8.4% thick symmetrical Von Mises airfoil is used for this case-run where the airfoil performs a 0.1 rad (or 5.73°) step change in AOA. Figure 5.1 illustrates the changes in the pressure distributions over the airfoil at time instances corresponding to the airfoil having traveled distances, in terms of chord length, of 0.2, 0.5, 1.0, 2.0 and ∞. The associated trailing wake patterns at these time instances (less $t = \infty$) are shown in Figure 5.2. The time-dependent build-up of aerodynamic coefficients of lift, drag, pitching moment and the circulation strength over a computation period of two traveled chord length are shown in Figure 5.3. Notice that the lift, pitching moment and circulation results are normalised by the respective steady state values at the same AOA. The apparently large initial loading on the airfoil shown in Figure 5.3 correlates

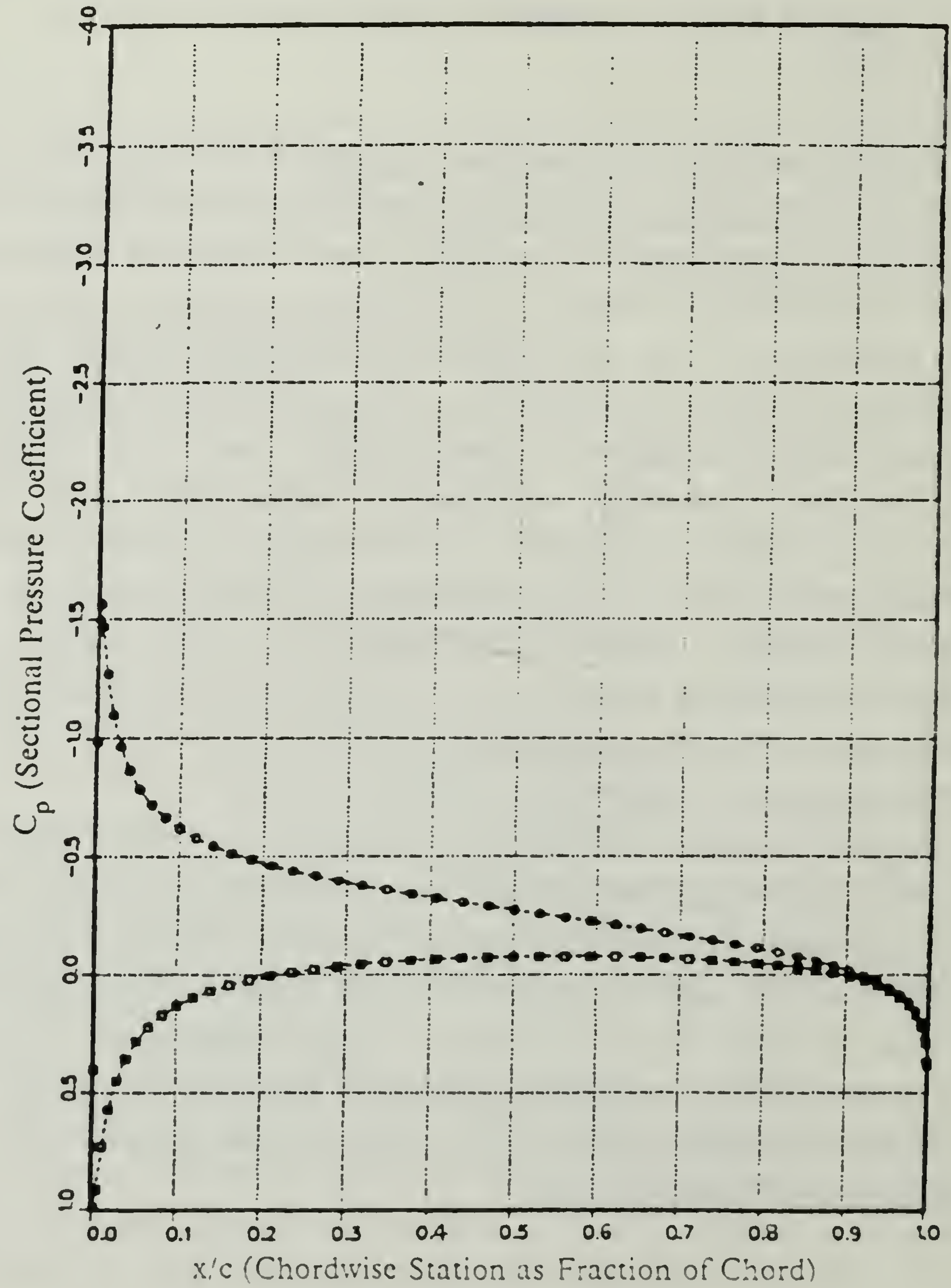
consistently with the results of the *Piston Theory* of [Ref. 6] which predicts the starting load on an arbitrary wing to be

$$C_{L_{starting}} = 4\alpha/M$$

where M is the Mach number. In the case of an incompressible flow ($M = 0$), the initial loading would be infinitely large. The same large initial loading was obtained by Kim and Mook [Ref. 7] who used continuous vorticities as panel singularities instead of our source and vorticity approach. Perhaps what remains most surprising is that the work of Basu and Hancock [Ref. 3] did not predict this initial loading, although they used the same singularity distributions as U2DIIF code. The initial large loading in lift and pitching moment decreases rapidly over a short time span, whereby the airfoil traveled approximately one-tenth chord length, before rising in a manner parallel to the Wagner Function. The drag, however, continues to decrease monotonically after the initial sharp fall. The circulation rises, as continuous shedding of vorticity takes place, slowly from the initial condition of zero to the asymptotic steady state value as time approaches ∞. These results, disregarding the initial large loading associated with incompressible flow, are in close agreement with the results of [Refs. 3,4,7].

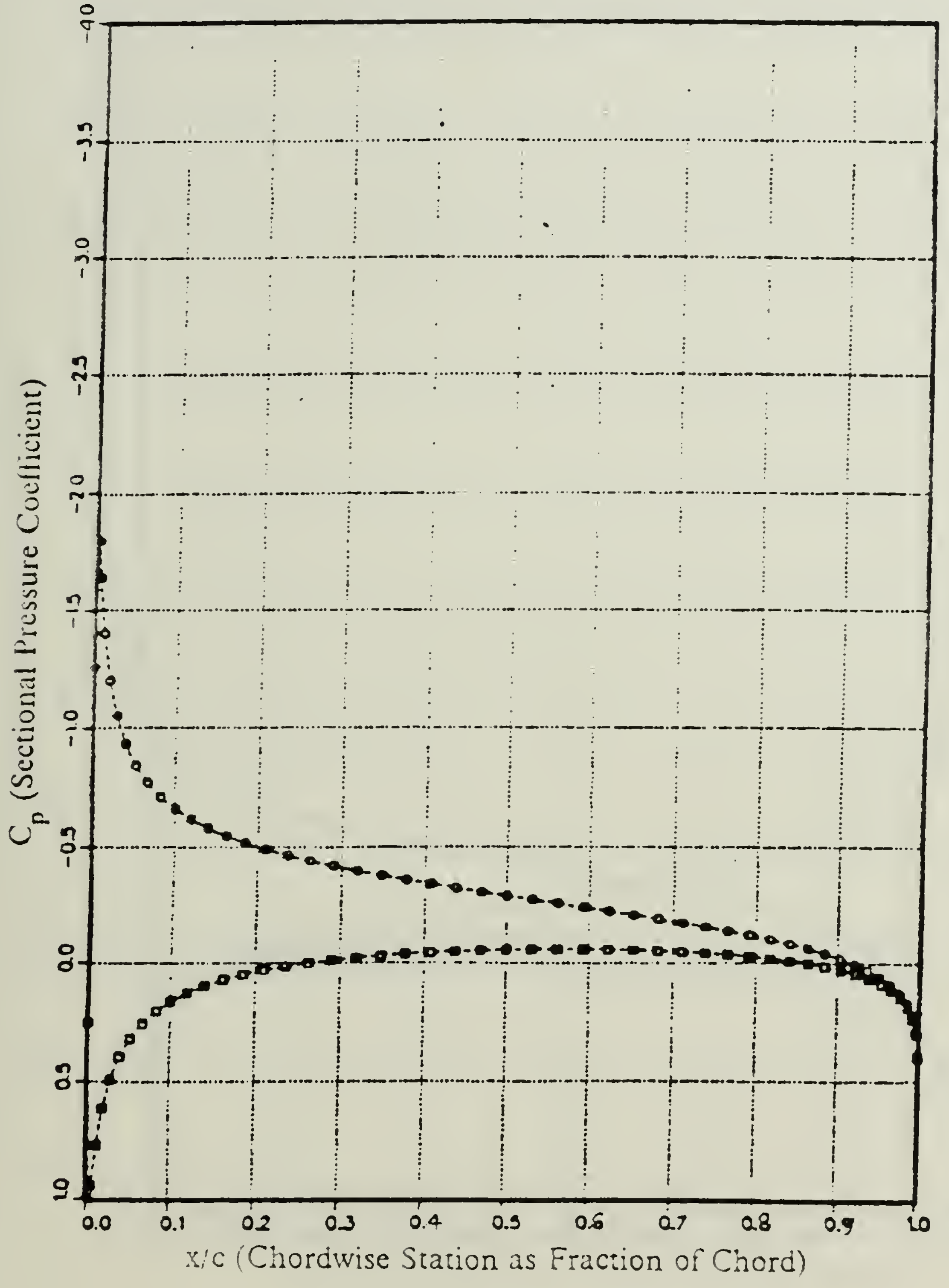
b. Thickness Effects on the Wagner Function

In order to more closely correlate the results of U2DIIF code to the theoretical prediction of Wagner, we performed the step AOA change calculations for a very thin (1% thickness) NACA 4-digit symmetrical airfoil which in reality should represent a flat plate. The results are plotted as shown in Figure 5.4. Shown also on the same Figure are the results of the 8.4% thick Von Mises airfoil and a 25.5% thick symmetric Joukowski airfoil. The initial loading falls off less rapidly for the case of the simulated flat plate as compared to other thick airfoils but the subsequent rise in lift follows very closely the Wagner Function.



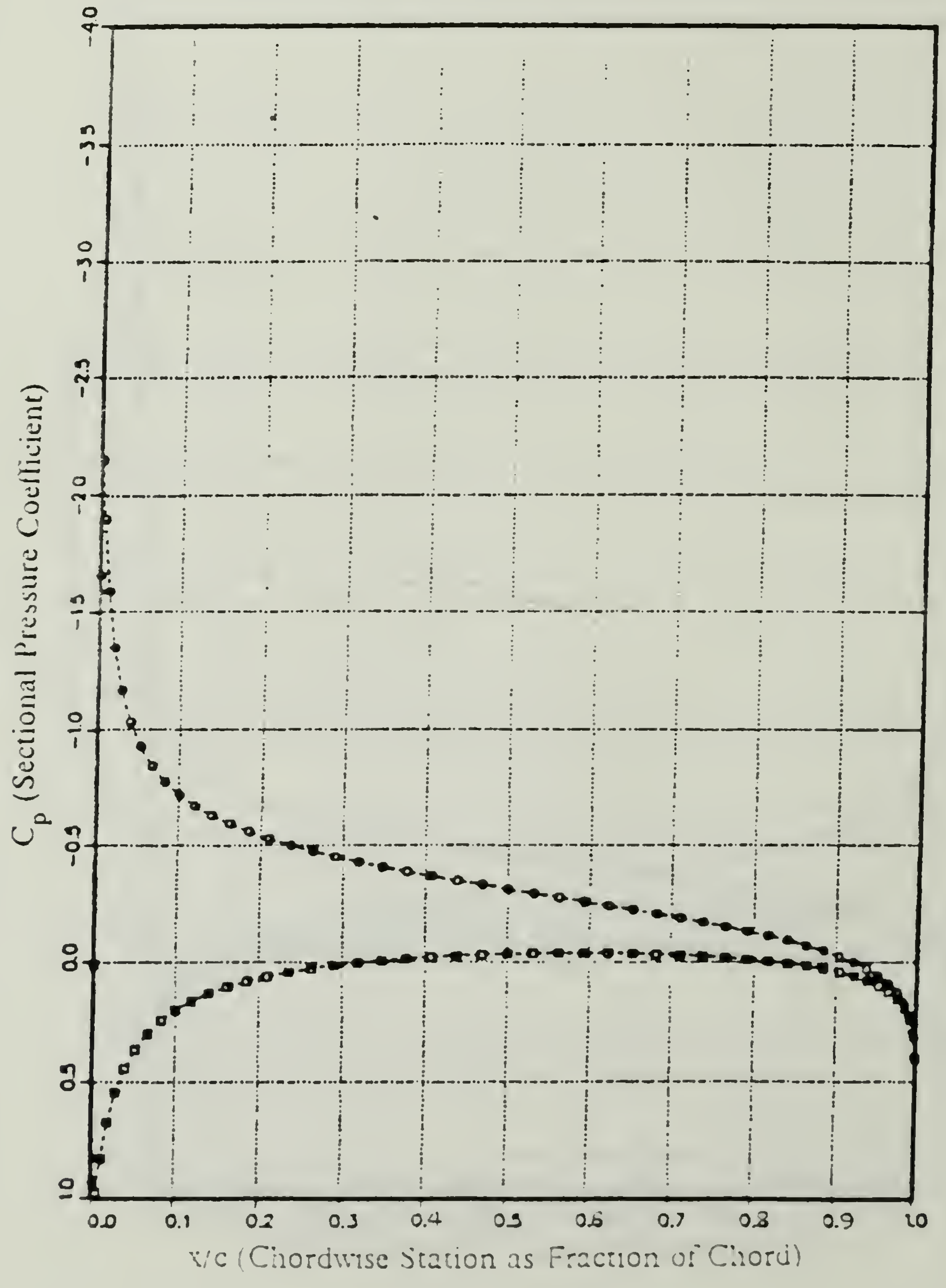
(a) $tV_\infty/c = 0.2$

Figure 5.1 Pressure Distributions at Various Time Instances Resulting from a 0.1 rad Step Change in AOA for a 8.4% Thick Symmetric Von Mises Airfoil.



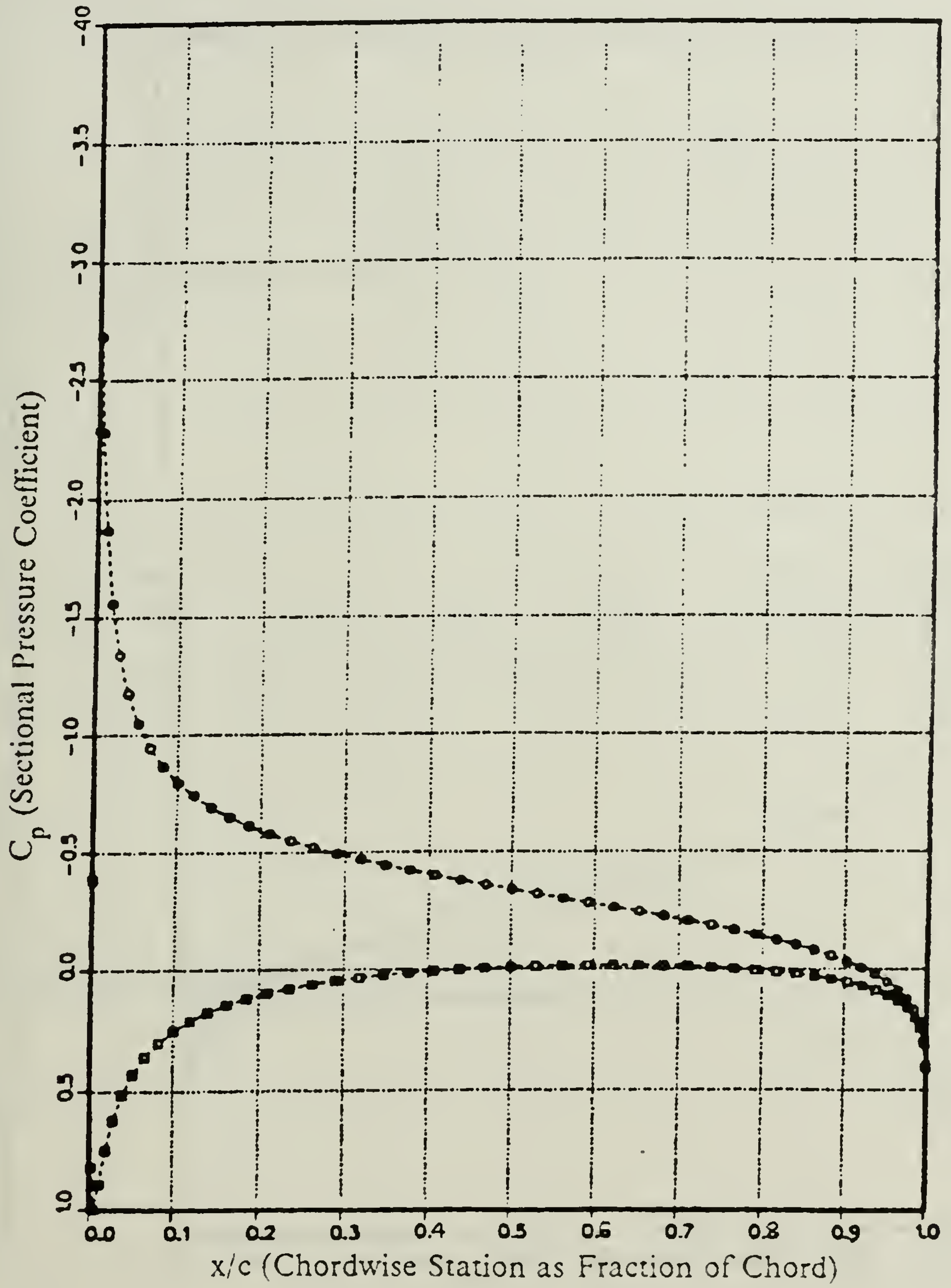
(b) $tV_{\infty}/c = 0.5$

Figure 5.1 . (cont'd.)



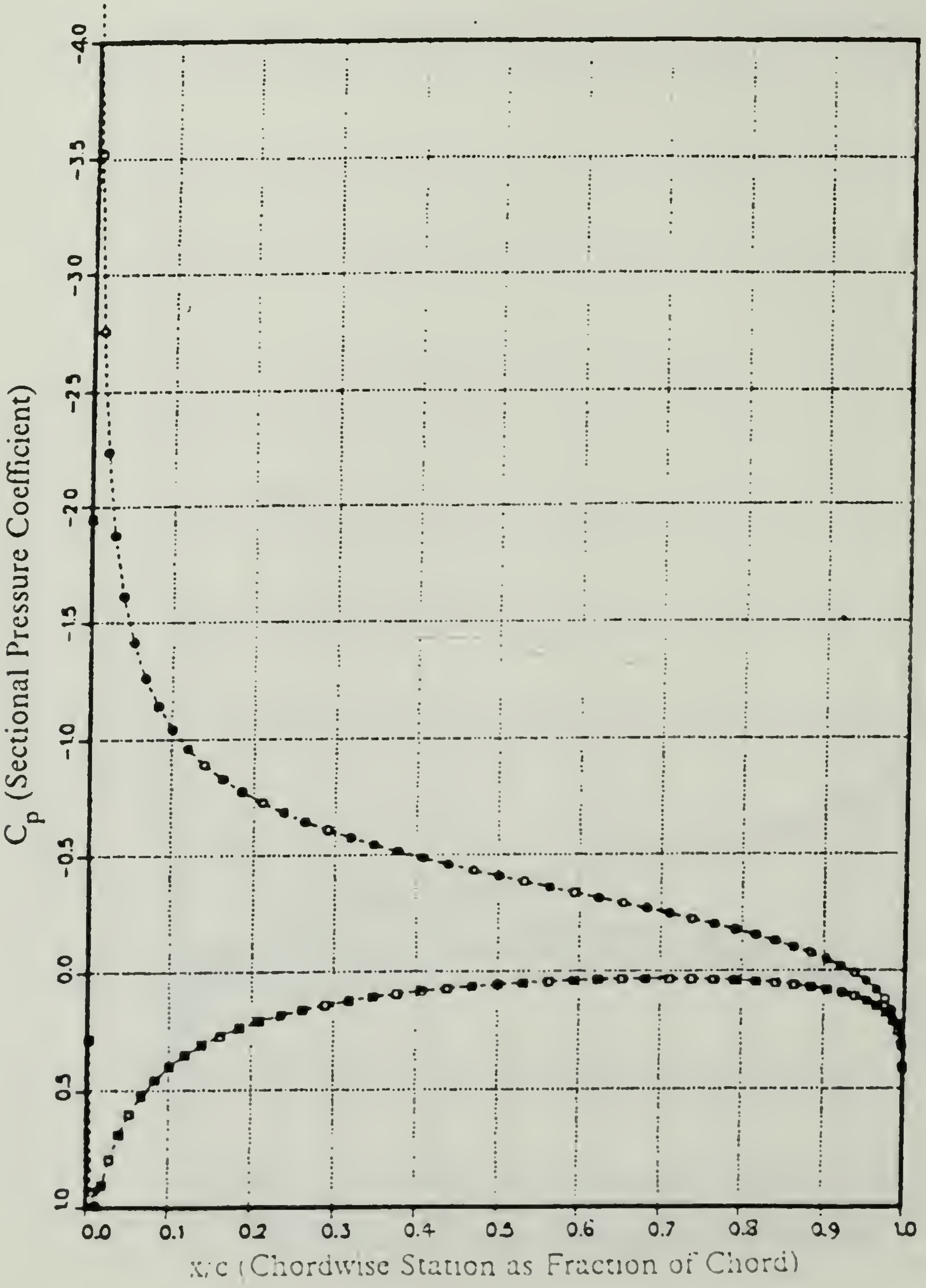
(c) $tV_\infty/c = 1.0$

Figure 5.1 . (cont'd.)



(d) $tV_{\infty}/c = 2.0$

Figure 5.1 . (cont'd.)



(e) $tV_{\infty}/c = \infty$

Figure 5.1 . (cont'd.)

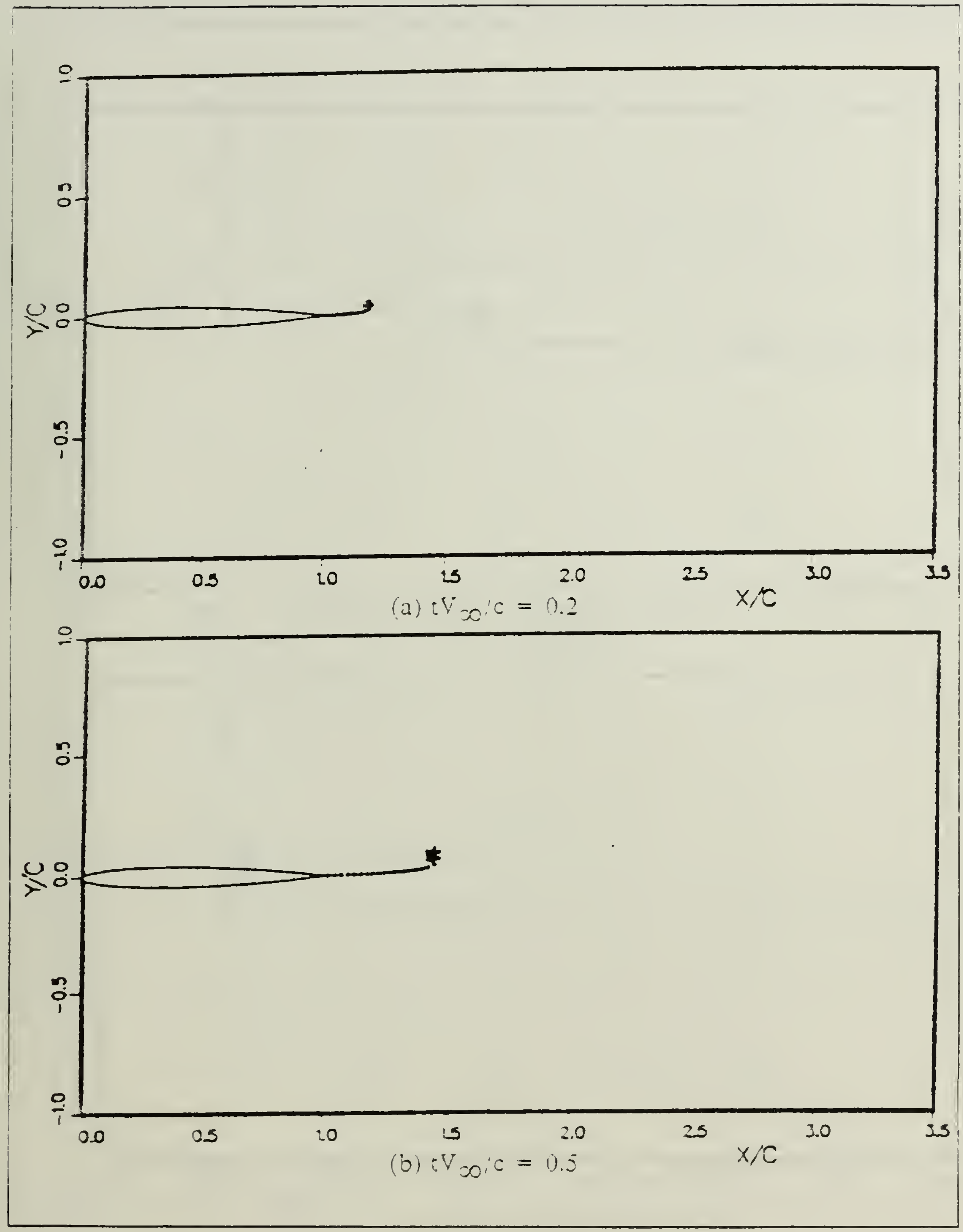
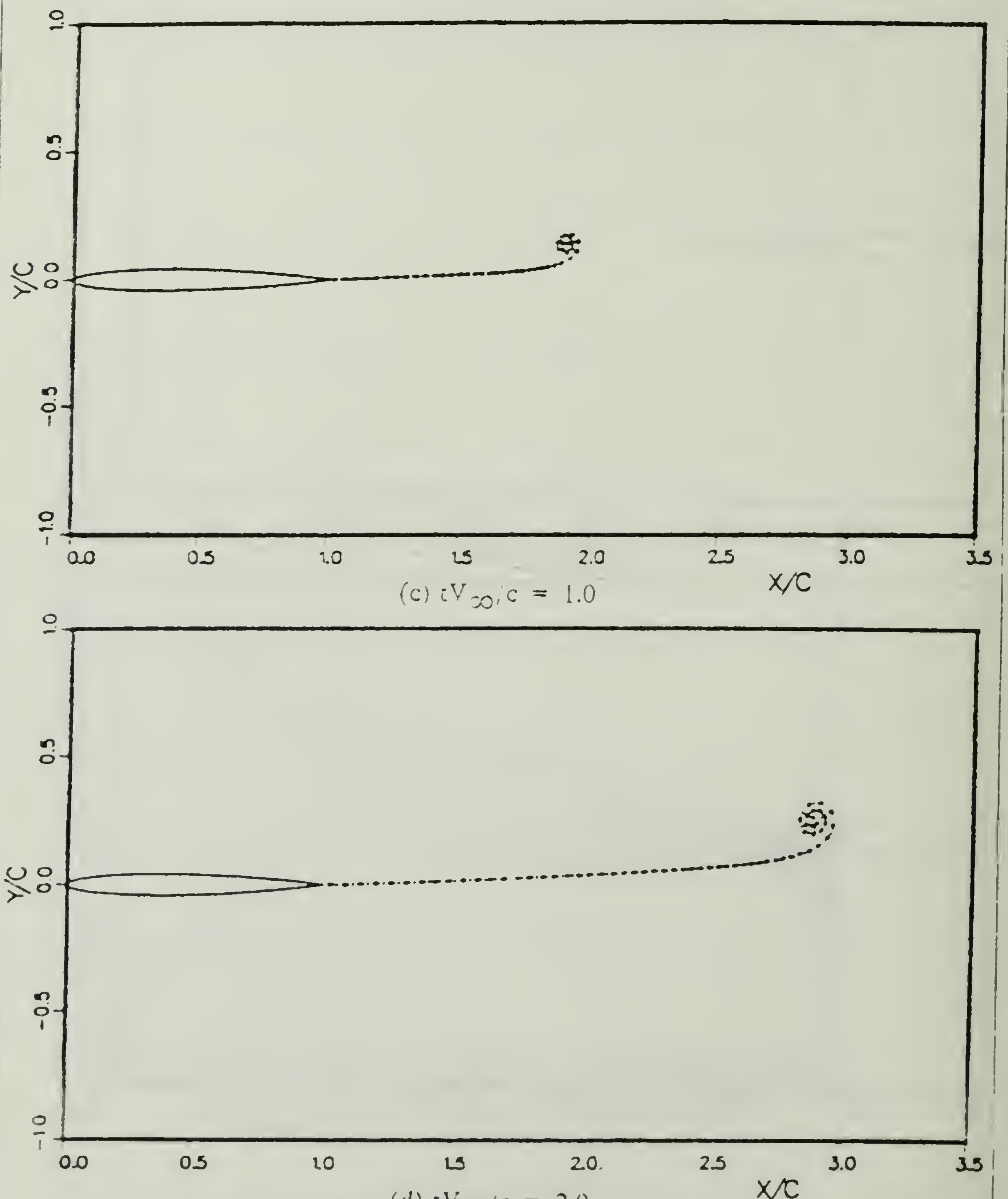


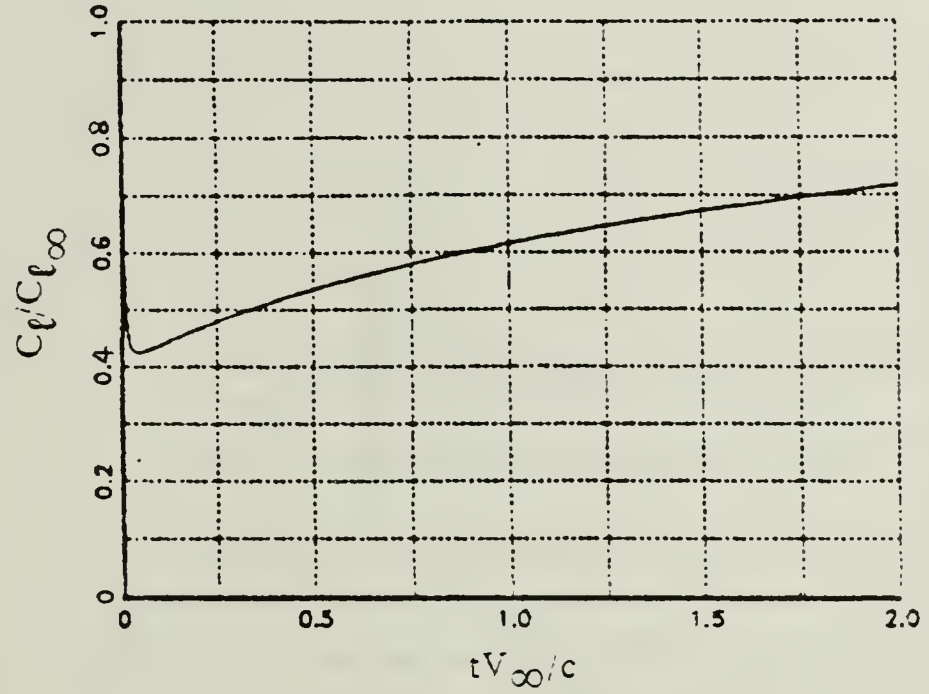
Figure 5.2 Trailing Wake Patterns at Various Time Instances Resulting from a 0.1 rad Step Change in AOA for a 8.4% Thick Symmetric Von Mises Airfoil.



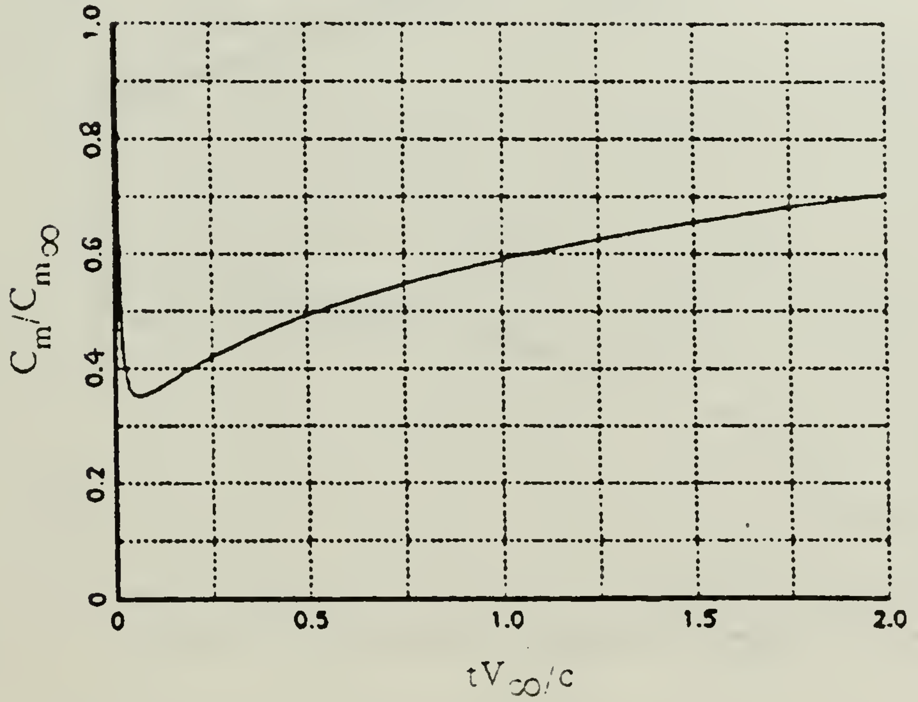
(c) $tV_{\infty}/c = 1.0$

(d) $tV_{\infty}/c = 2.0$

Figure 5.2 . (cont'd.)

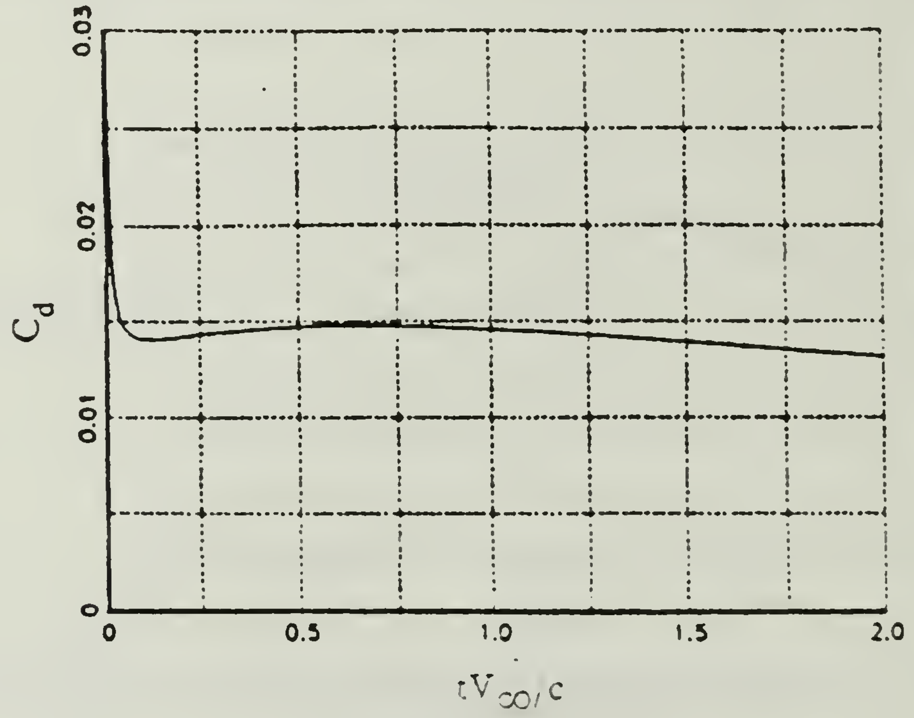


(a) Normalised Lift C_ℓ/C_{ℓ_∞} vs tV_∞/c.

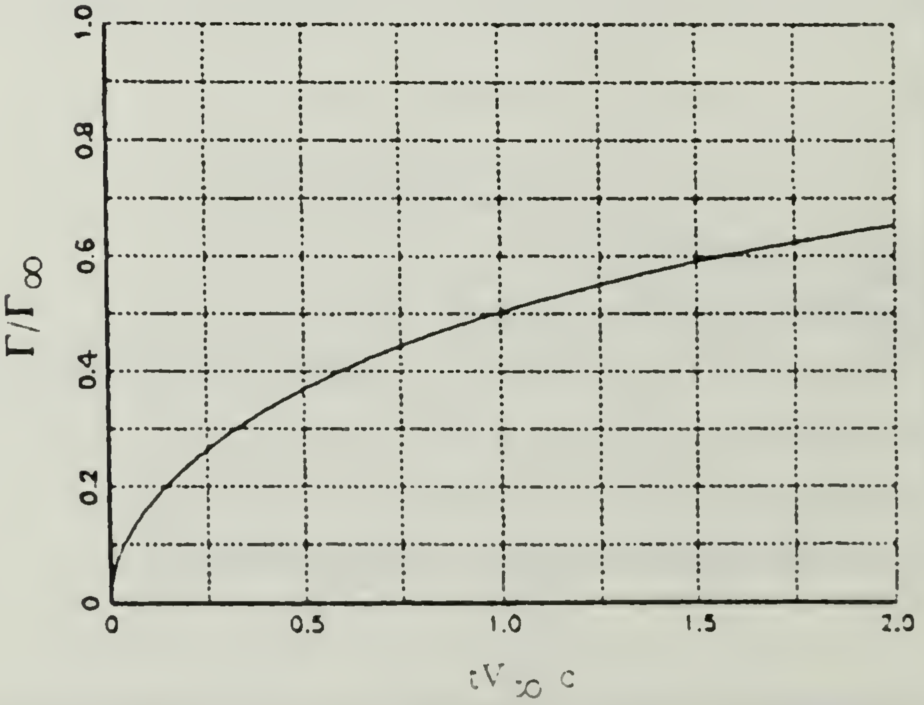


(b) Normalised Pitching Moment C_m/C_{m_∞} vs tV_∞/c.

Figure 5.3 Time-Dependent Aerodynamic Parameters Resulting from a 0.1 rad Step Change in AOA for a 8.4% Thick Symmetric Von Mises Airfoil.



(c) Drag C_d vs tV_∞/c.



(d) Normalised Circulation Γ/Γ_∞ vs tV_∞/c.

Figure 5.3 . (cont'd.)

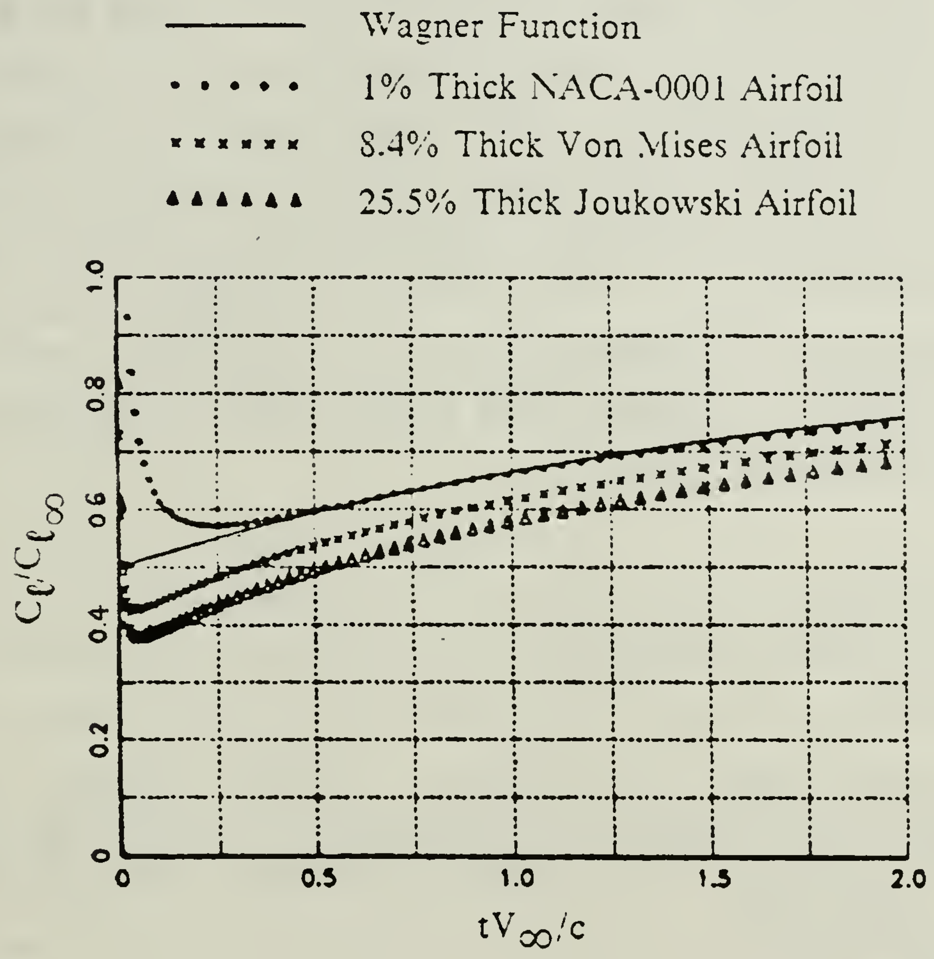


Figure 5.4 Time-Dependent Lift Resulting from Step Change in AOA for Airfoils of Various Thicknesses.

B. MODIFIED RAMP CHANGE IN ANGLE-OF-ATTACK

1. Case-Run Definitions

The case of a step change in AOA can be considered as an useful check for U2DIIF code since a handful of results from other theoretical analyses are available. However, a step change in AOA is practically not realisable since all motions, short of having infinite velocities, take place over a finite time span. The step change in AOA that is practically possible is in fact some form of ramp rise over a short time span with large velocity. Even so, due to the inertia of the airfoil, an exact ramp rise in AOA does not physically describe the actual motion of the airfoil since finite time is also involved before the airfoil could build up its ramp velocity. Same argument holds at the end of the ramp rise before the airfoil could stop at the final value of AOA. Therefore a so called *modified ramp*, with some form of *rounding* at the two ends of a ramp, is more likely to describe anything close to what is physically achievable. The theoretical work of Homentcovschi in [Ref. 8] considers the case of a flat plate that moves with constant velocity and changes the incidence about the mid chord, through a particular ramp fashion, described mathematically as,

$$\alpha(t) = \begin{cases} 0 & \text{for } t < 0, \\ \delta\alpha\,(3 - 2t/\tau)\, t^2/\tau^2 & \text{for } 0 \le t \le \tau \\ \delta\alpha & \text{for } t > \tau \end{cases}$$

where $\delta\alpha$ is the magnitude of the AOA change and τ is the rise time for the AOA to reach its final value. This particular function, plotted as shown in Figure 5.5, does in fact describe such a modified ramp.

2. Results and Discussions

a. Flat-Plate Case-Run

Since the results of [Ref. 8] serves as another excellent source for the verification of U2DIIF code, the obvious thing to do is to use U2DIIF to compute for the case of a flat plate, again simulated by the 1% thick NACA-0001 airfoil, executing this modified ramp rise of 0.1 rad AOA over a rise time of 1.5 chord length. This rise time is chosen simply to facilitate a direct comparison of results to [Ref. 8] which used

a rise time of 3 half-chord lengths. The results of computation are shown in Figure 5.6 and 5.7. Figure 5.6(a) takes a close look at the build up of lift during the initial period when the airfoil moves a distance of six chord lengths. The lift initially rises to about 82% and then decreases to about 66% of the steady state value during the transient rise time. Thereafter it increases monotonically in a manner parallel to the Wagner Function. Figure 5.6(b) is a zoom view of the rather slow convergence of lift to the steady state value. It takes the airfoil to cover a distance of around 50 chord length before the lift builds up to almost 99% of the steady state value. The same results were obtained in the theoretical analysis of [Ref. 8]. Figure 5.7 shows a collection of the time-dependent aerodynamic parameters resulting from this particular case-run.

b. Thickness Effects

The same modified ramp function is used on the 8.4% thick Von Mises airfoil. The resulting lift-history plotted in Figure 5.8 shows a lower peak value of lift during the transient AOA rise as compared to the case of a flate plate though a similar trend of lift rise is obtained. Figures 5.9 and 5.10 show the results of pressure distributions and trailing wake patterns at various time instances. One could directly compare these Figures to the corresponding Figures arising from the step AOA change calculations and see the remarkable differences in transient characteristics as a result of varying the prescribed motions. Incidentally, one should realise that the non-dimensional rise time of 1.5 chord length is a deceivingly large number. In fact, when one converts this to the real time for an airfoil of 10 ft chord length moving at a low Mach number of 0.2, the rise time is indeed only of the order of 0.06 sec. which for practical purpose is close enough to a step. Neverthless, the transient part of the lift response is entirely governed by how one prescribes the transient motion.

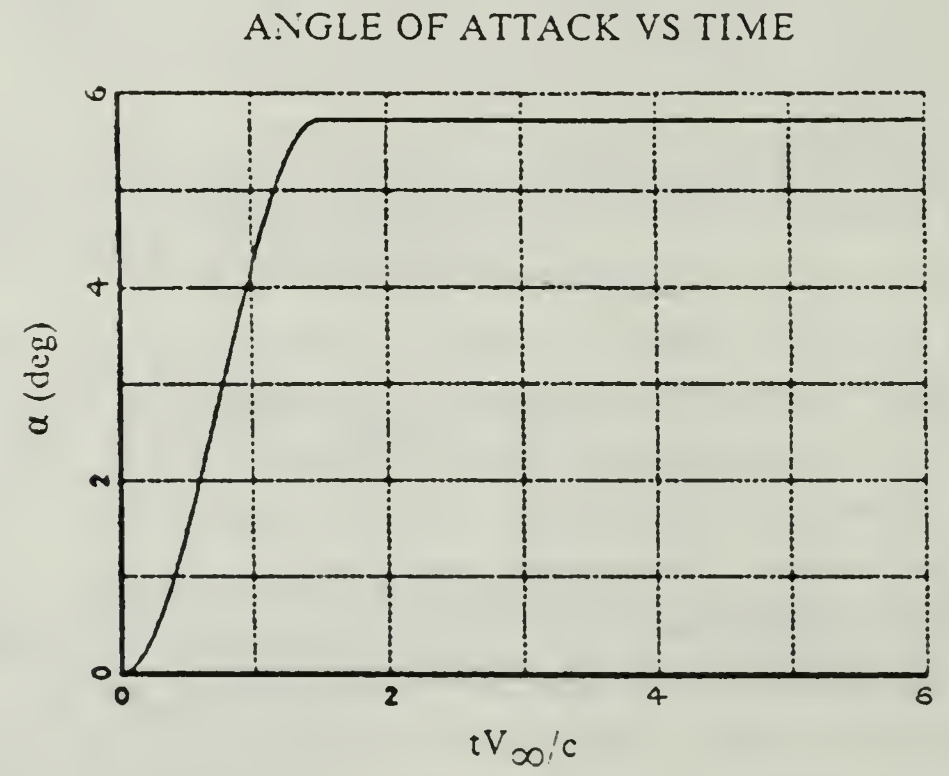
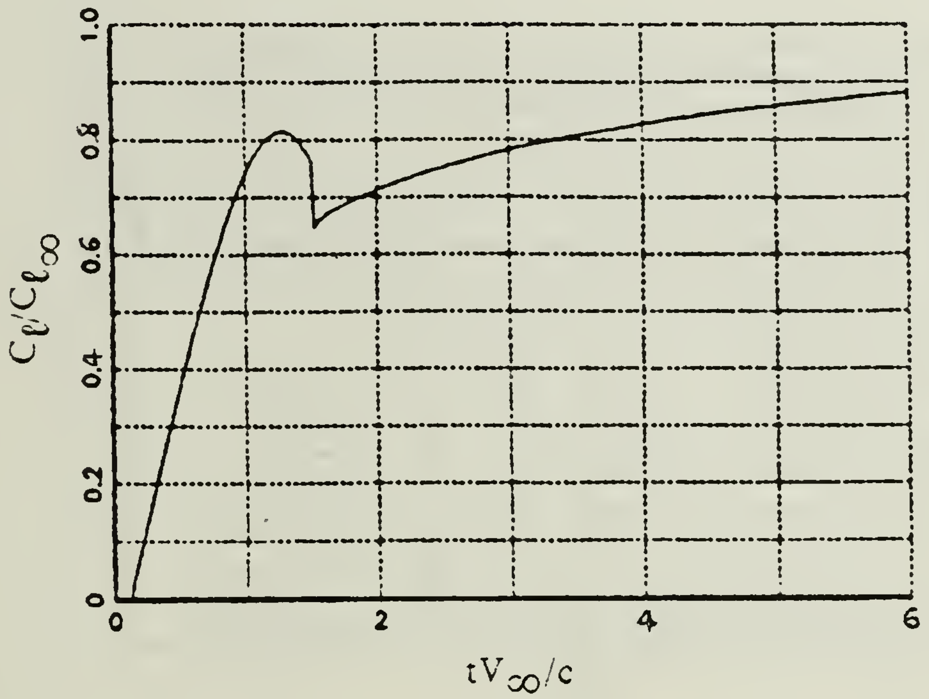
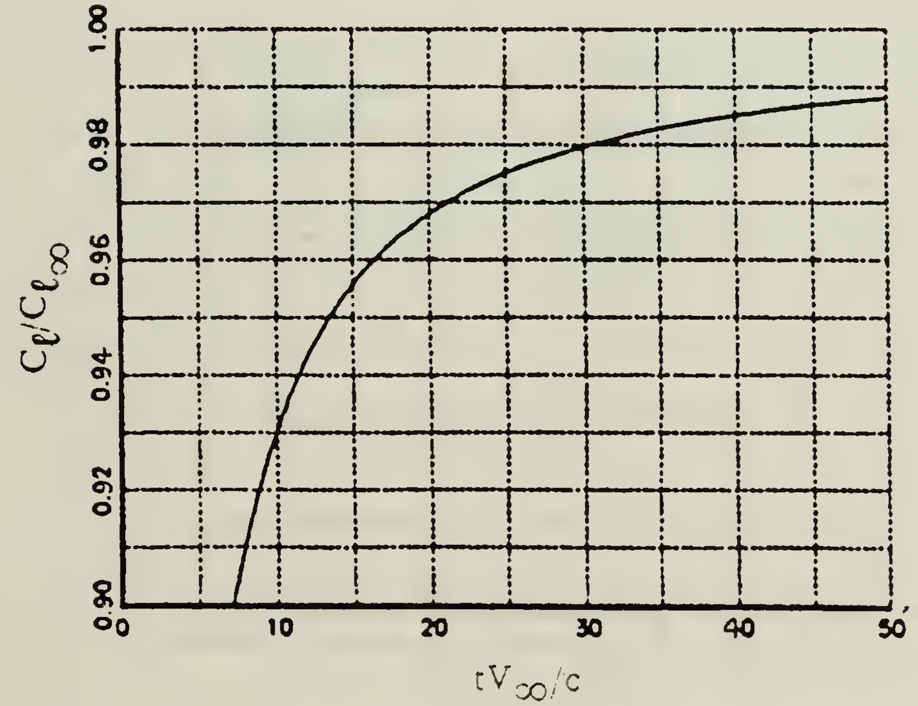


Figure 5.5 The Modified Ramp AOA Change.

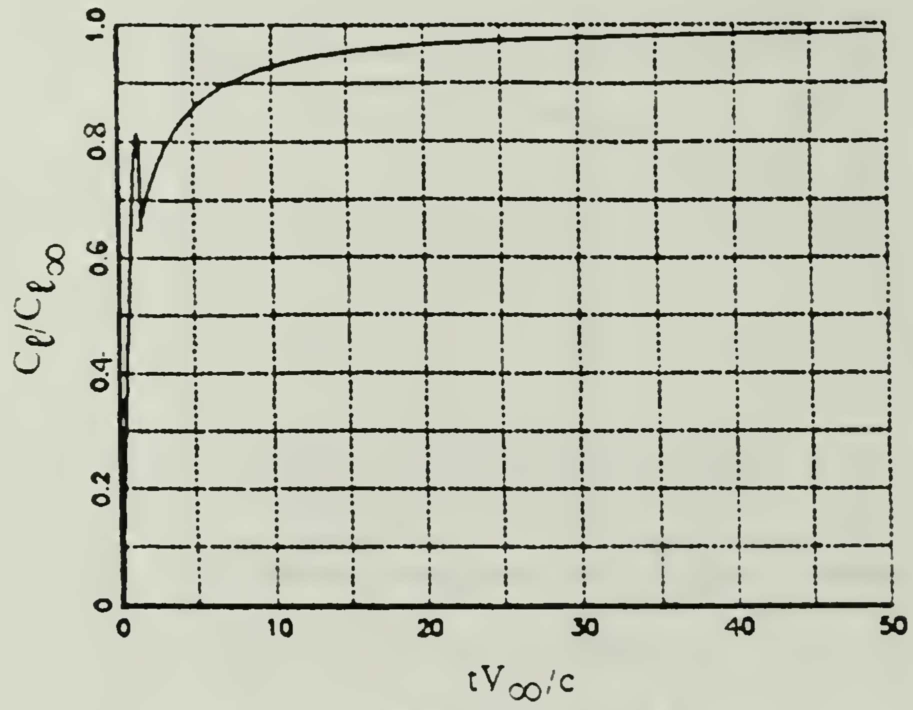


(a) $0 \leq tV_\infty/c \leq 6$

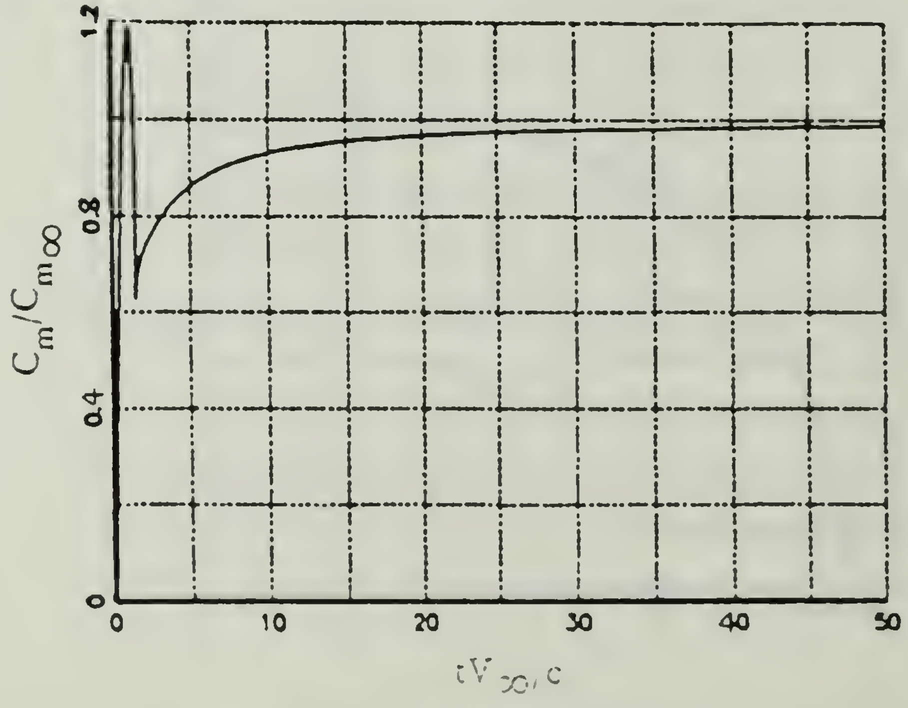


(b) $0 \leq tV_\infty/c \leq 50$

Figure 5.6 Normalised Lift C_ℓ/C_{ℓ_∞} Resulting from a Modified Ramp AOA ($\delta\alpha$ = 0.1 rad, τ = 1.5) about the Mid Chord of a NACA-0001 Airfoil.

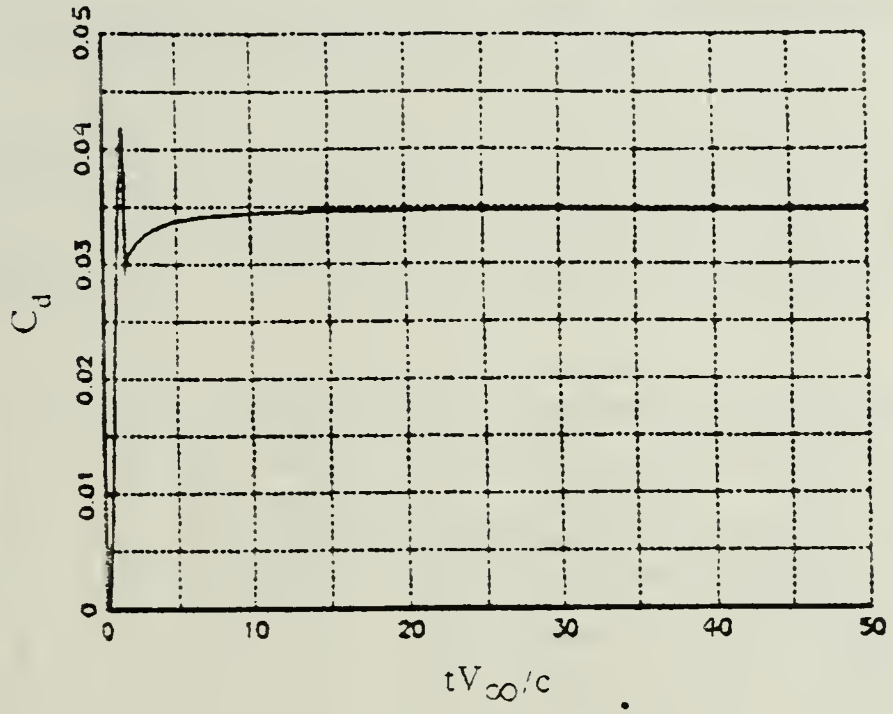


(a) Normalised Lift C_ℓ/C_{ℓ_∞} vs tV_∞/c.

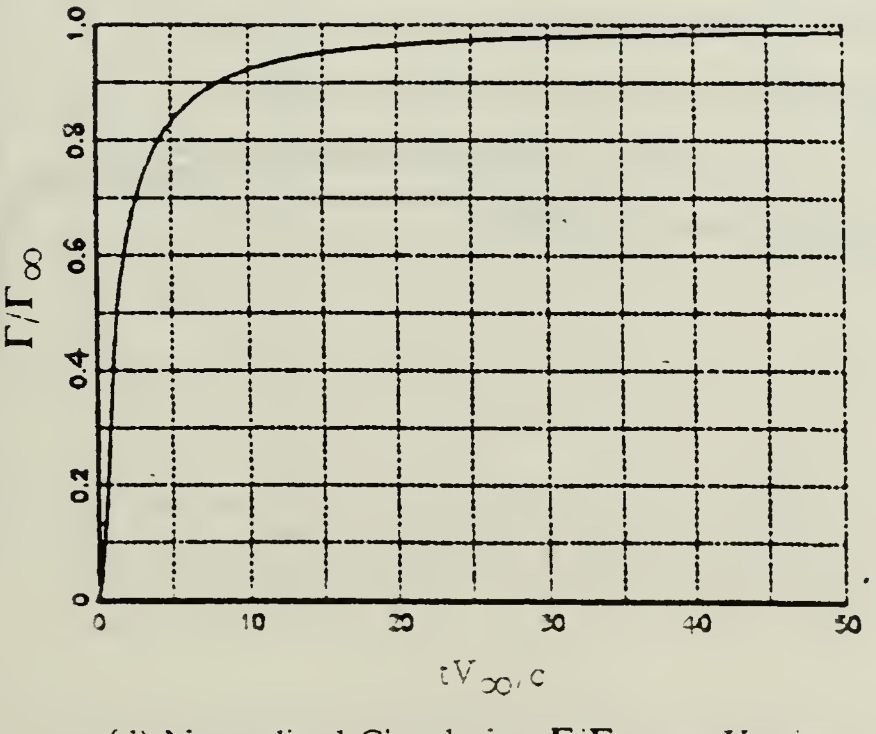


(b) Normalised Pitching Moment C_m/C_{m_∞} vs tV_∞/c.

Figure 5.7 Time-Dependent Aerodynamic Parameters Resulting from a Modified Ramp AOA ($\delta\alpha$ = 0.1 rad, τ = 1.5) about the Mid Chord of a NACA-0001 airfoil.



(c) Drag C_d vs tV_∞/c.



(d) Normalised Circulation $\mathbf{\Gamma/\Gamma_\infty}$ vs tV_∞/c.

Figure 5.7 . (cont'd.)

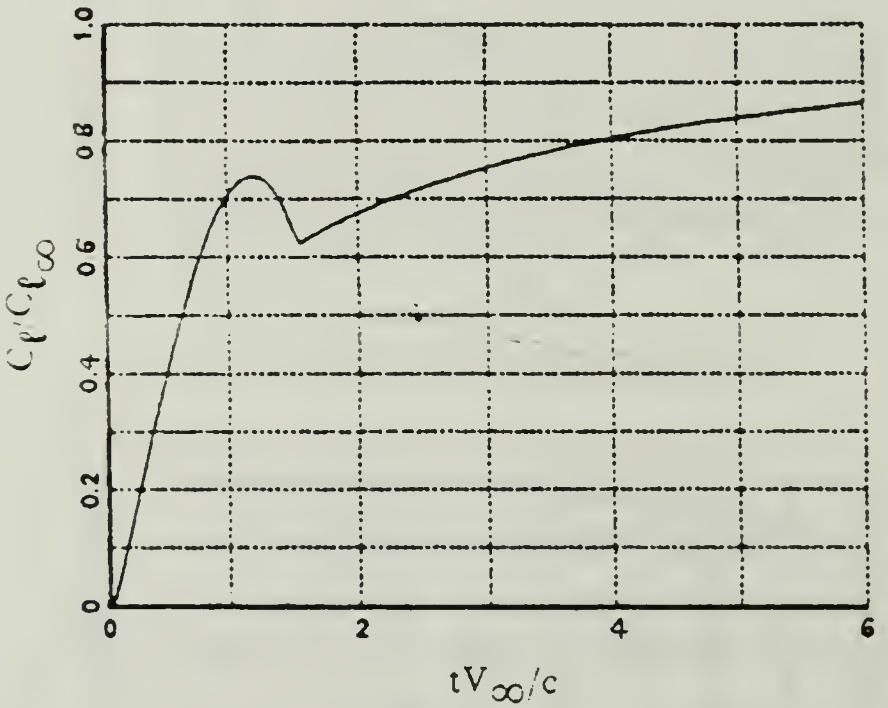
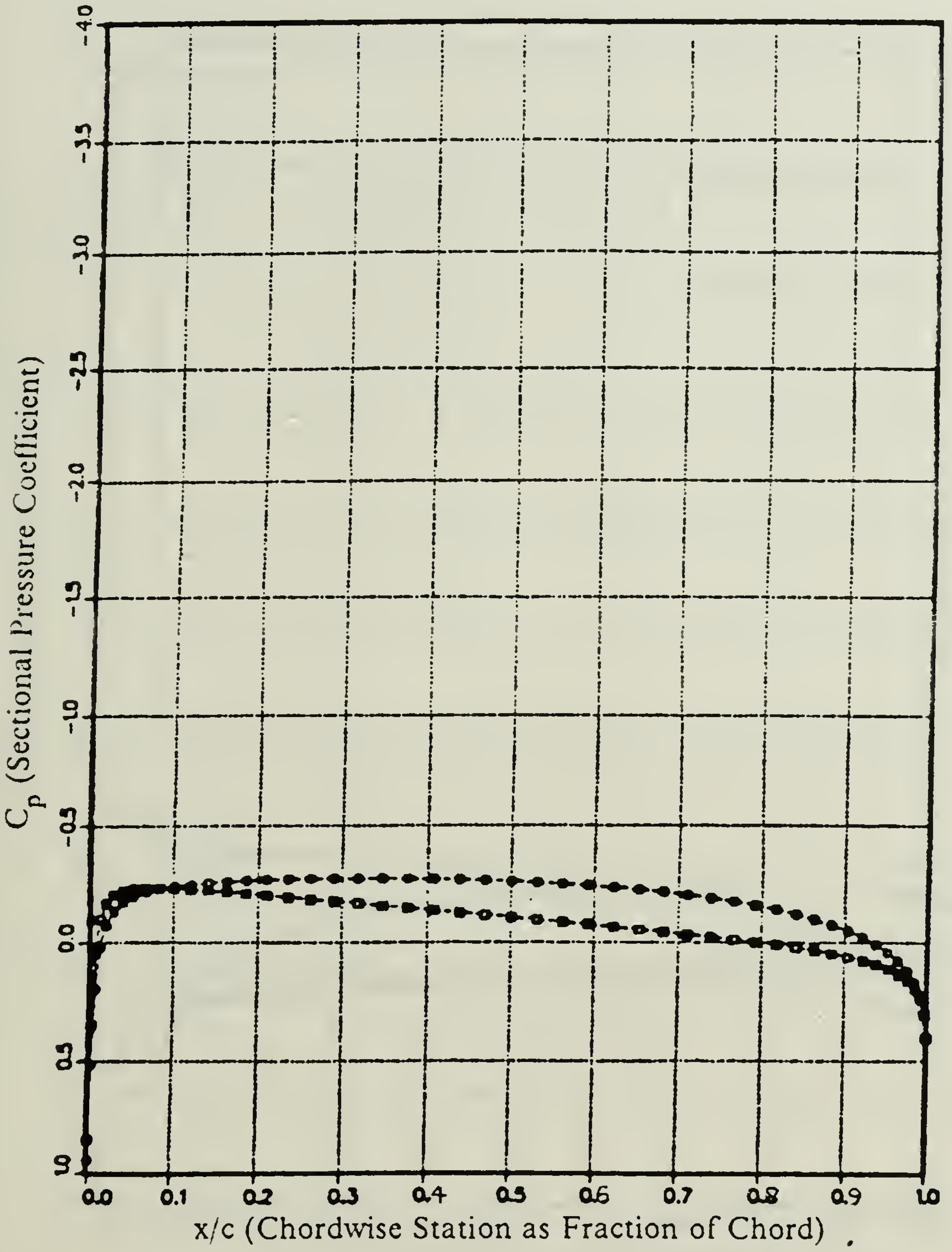
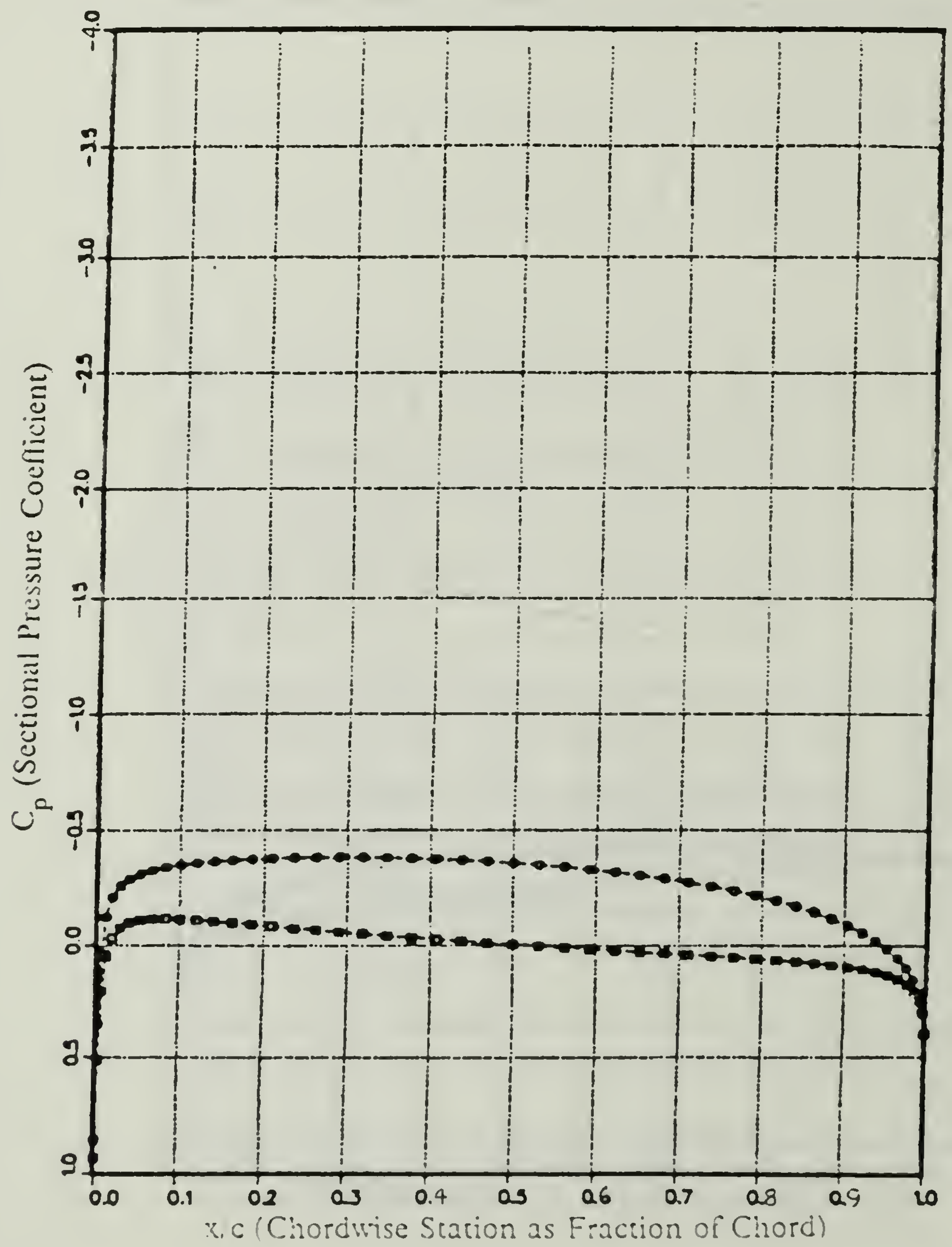


Figure 5.8 Normalised Lift C_ℓ / C_{ℓ_∞} Resulting from a Modified Ramp AOA ($\delta\alpha$ = 0.1 rad, τ = 1.5) about the Mid Chord of a Mises 8.4% Thick airfoil.



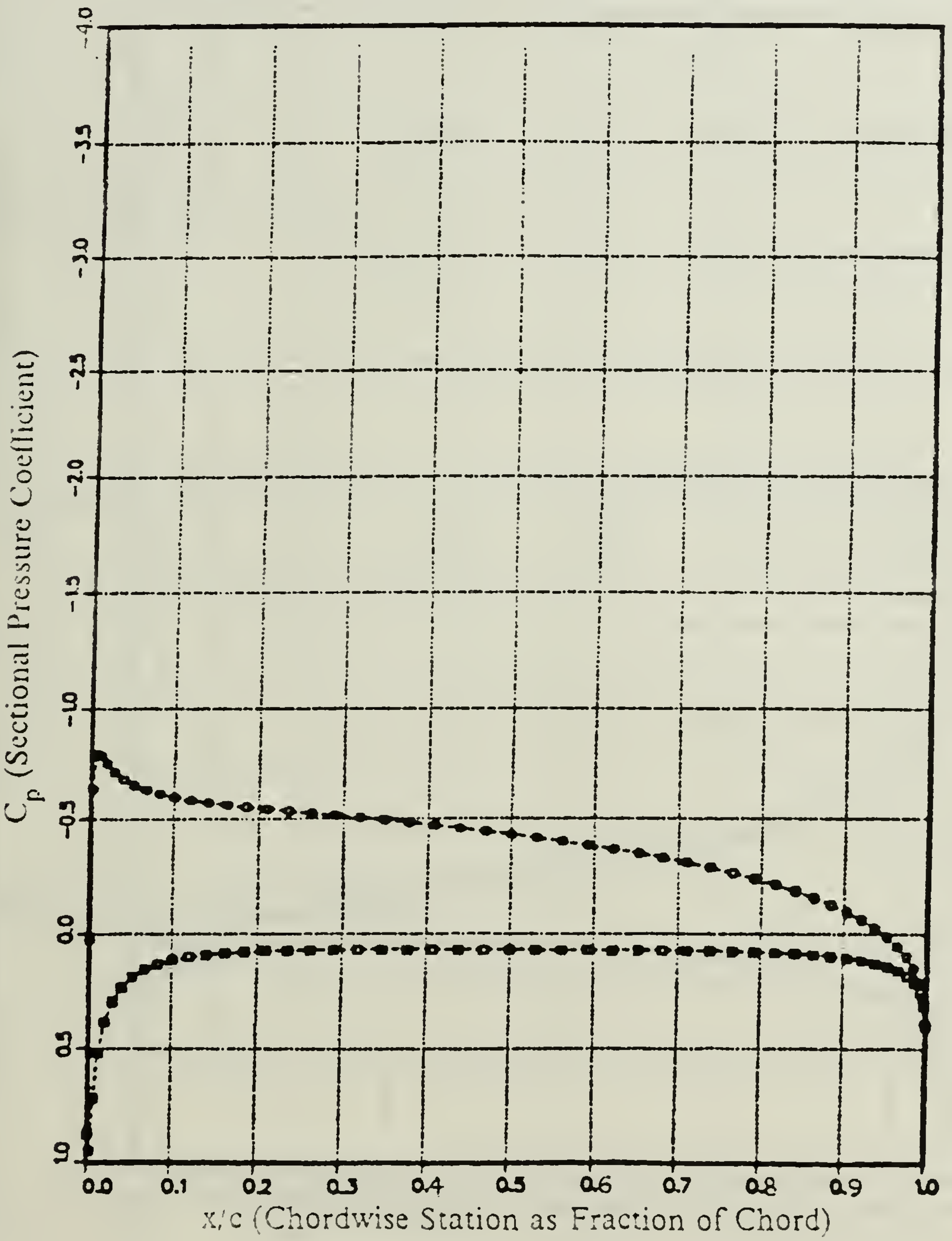
(a) $tV_{\infty}/c = 0.2$

Figure 5.9 Pressure Distributions at Various Time Instances Resulting from a Modified Ramp AOA ($\delta\alpha = 0.1$ rad, $\tau = 1.5$) about the Mid Chord of a Mises 8.4% Thick airfoil.



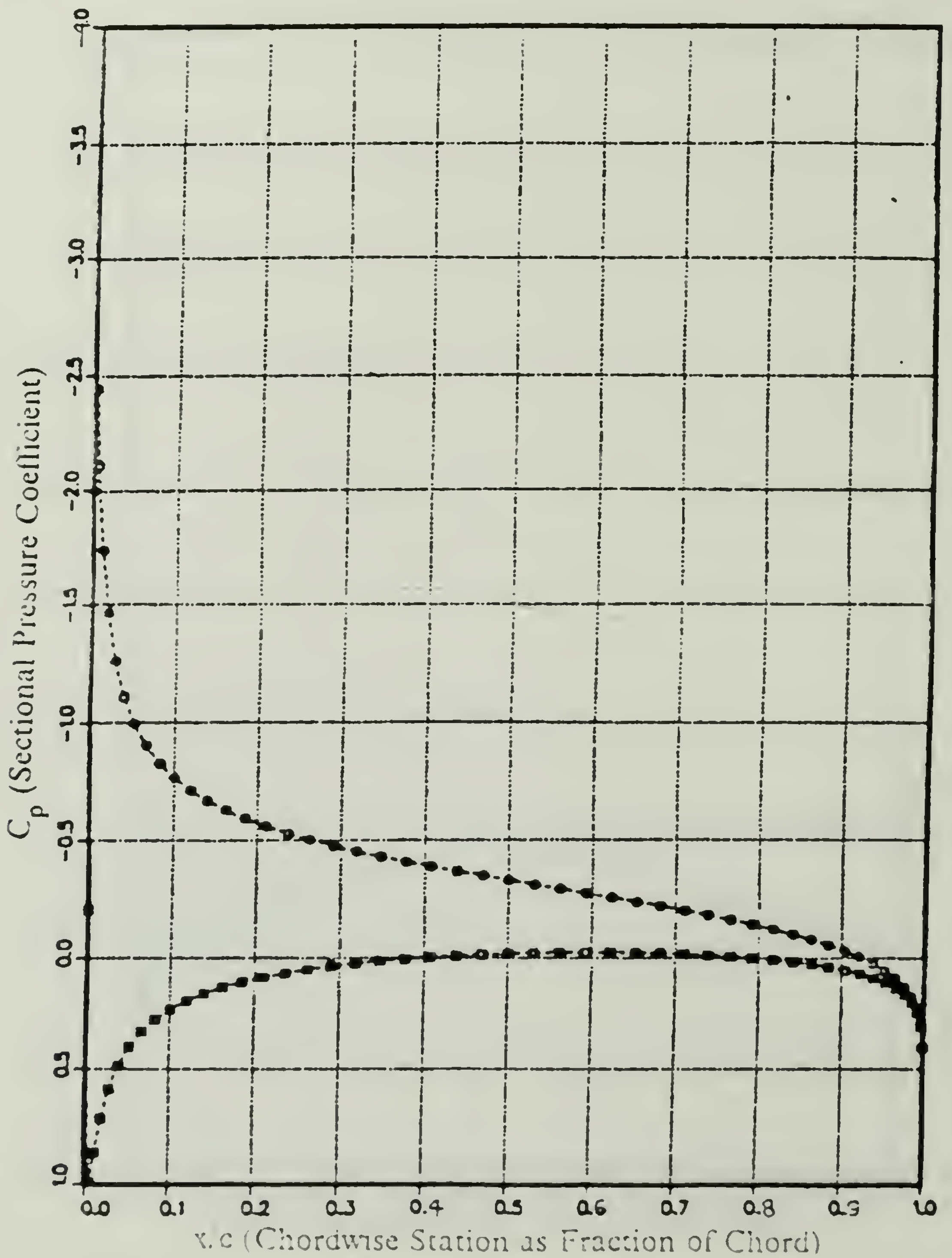
(b) $tV_\infty/c = 0.5$

Figure 5.9 . (cont'd.)



(c) $tV_{\infty}/c = 1.0$

Figure 5.9 . (cont'd.)



(d) $tV_\infty/c = 2.0$

Figure 5.9 . (cont'd.)

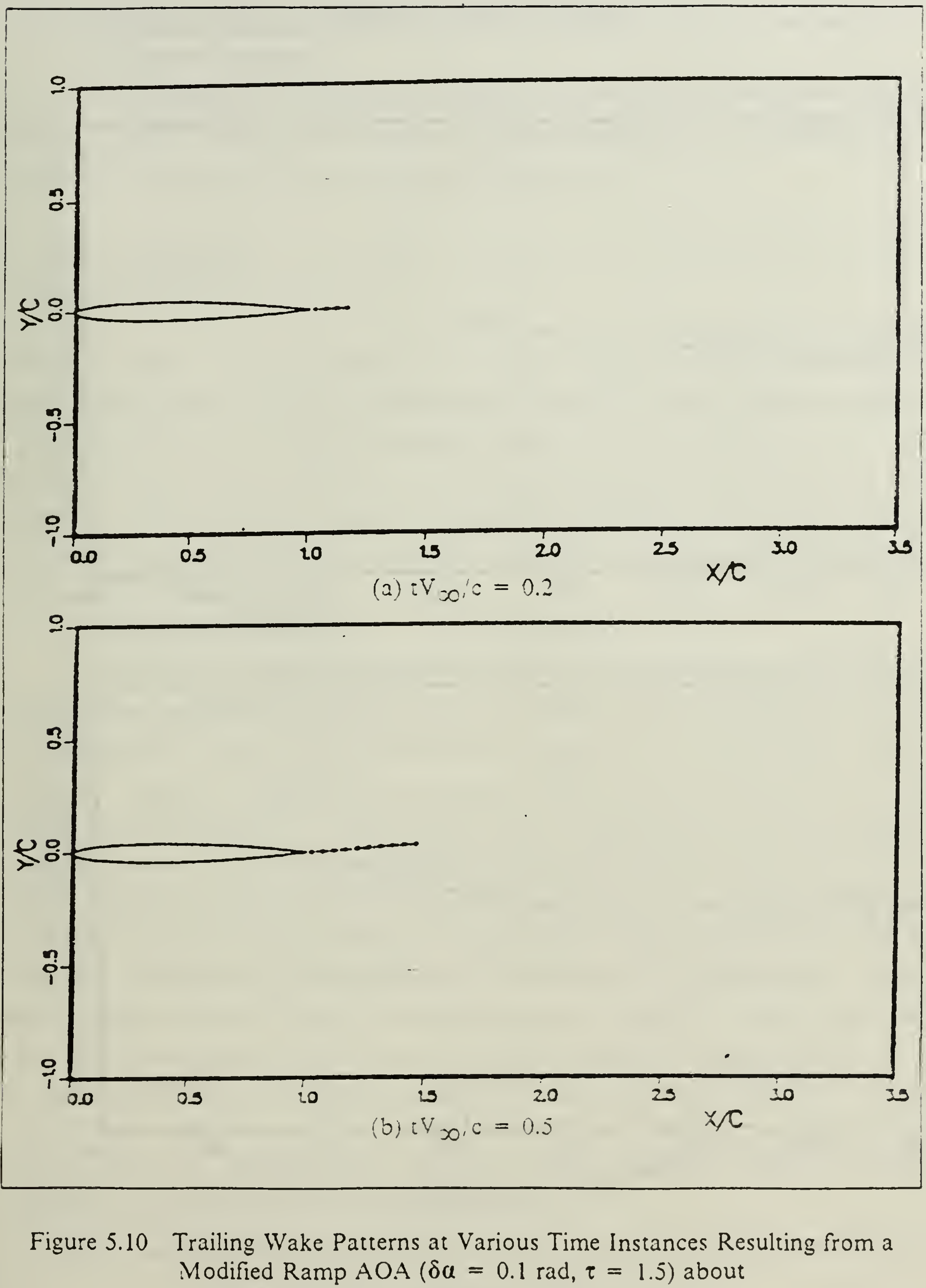
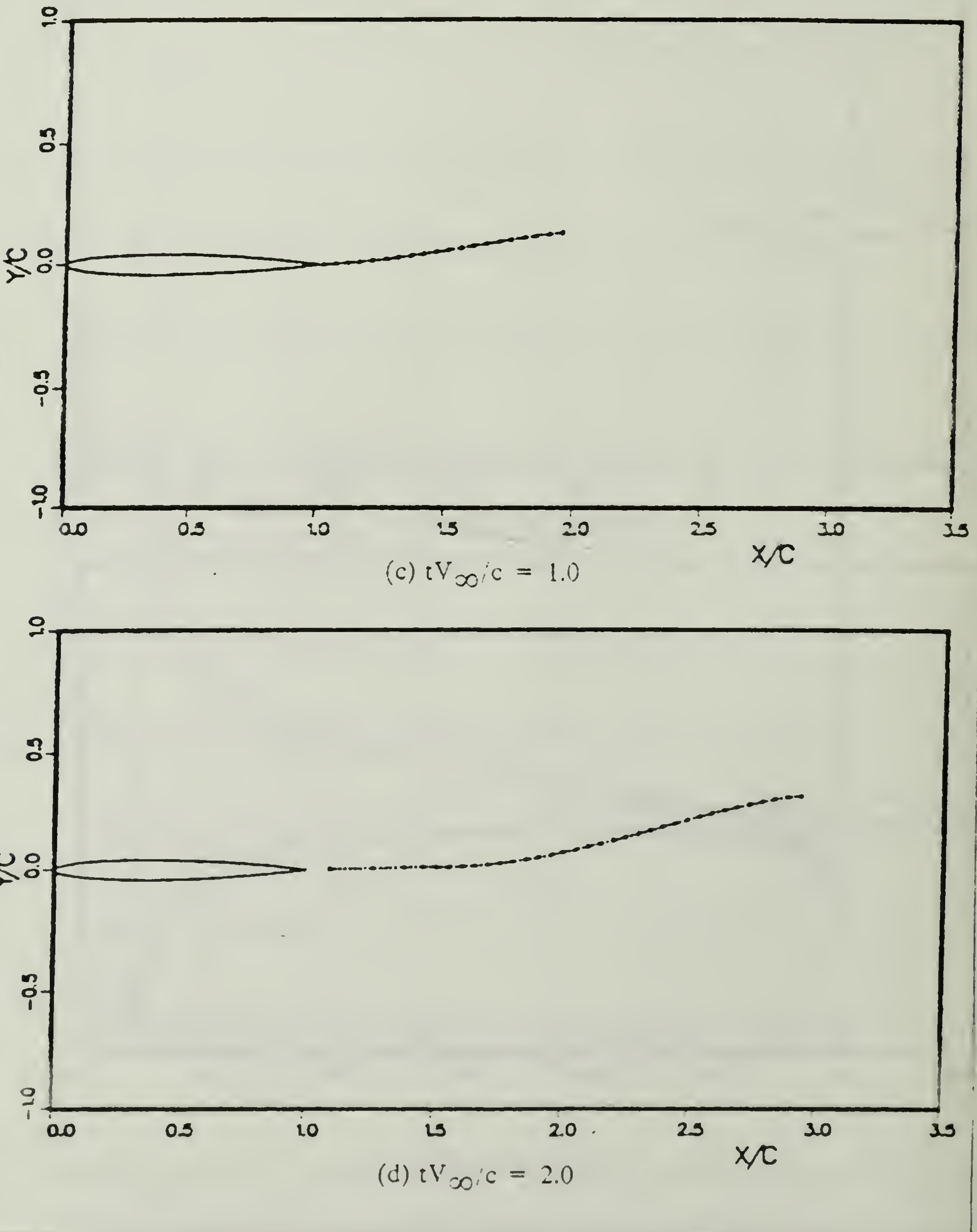


Figure 5.10 Trailing Wake Patterns at Various Time Instances Resulting from a Modified Ramp AOA ($\delta\alpha$ = 0.1 rad, τ = 1.5) about the Mid Chord of a Mises 8.4% Thick airfoil.



(c) $tV_\infty/c = 1.0$

(d) $tV_\infty/c = 2.0$

Figure 5.10 . (cont'd.)

C. TRANSLATIONAL HARMONIC OSCILLATION

1. Case-Run Definitions

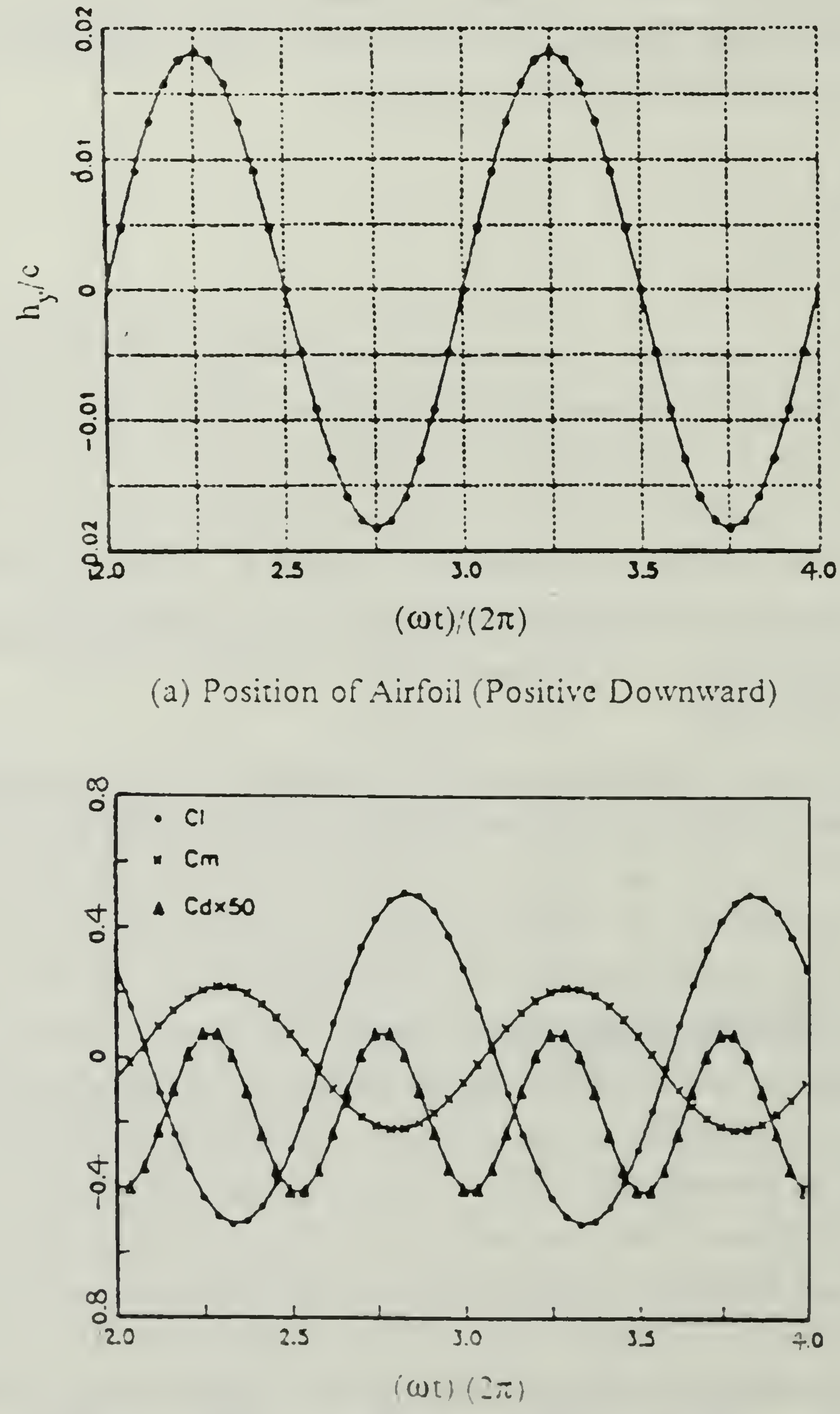
Although the U2DIIF code is capable of computing unsteady flow solution for any general translational harmonic motion described by a chordwise and a transverse components bearing a given phase relationship,

$$h_y(t) = \delta h_y \sin(\omega t)$$
$$h_x(t) = \delta h_x \sin(\omega t + \lambda)$$

where ω is the oscillation frequency, λ is the phase angle between the two oscillation components and δh_x & δh_y are the magnitudes of chordwise and transverse oscillations respectively. The case-run to be presented in this section selects the motion to consist of only the transverse component, i.e. the heaving or plunging motion. A NACA-0015 airfoil is chosen for the case-run. The airfoil is initially at zero AOA with the freestream V_∞ and performs the plunging oscillation at an amplitude of δh_y and a reduced frequency of $\omega c/V_\infty$

2. Results and Discussions

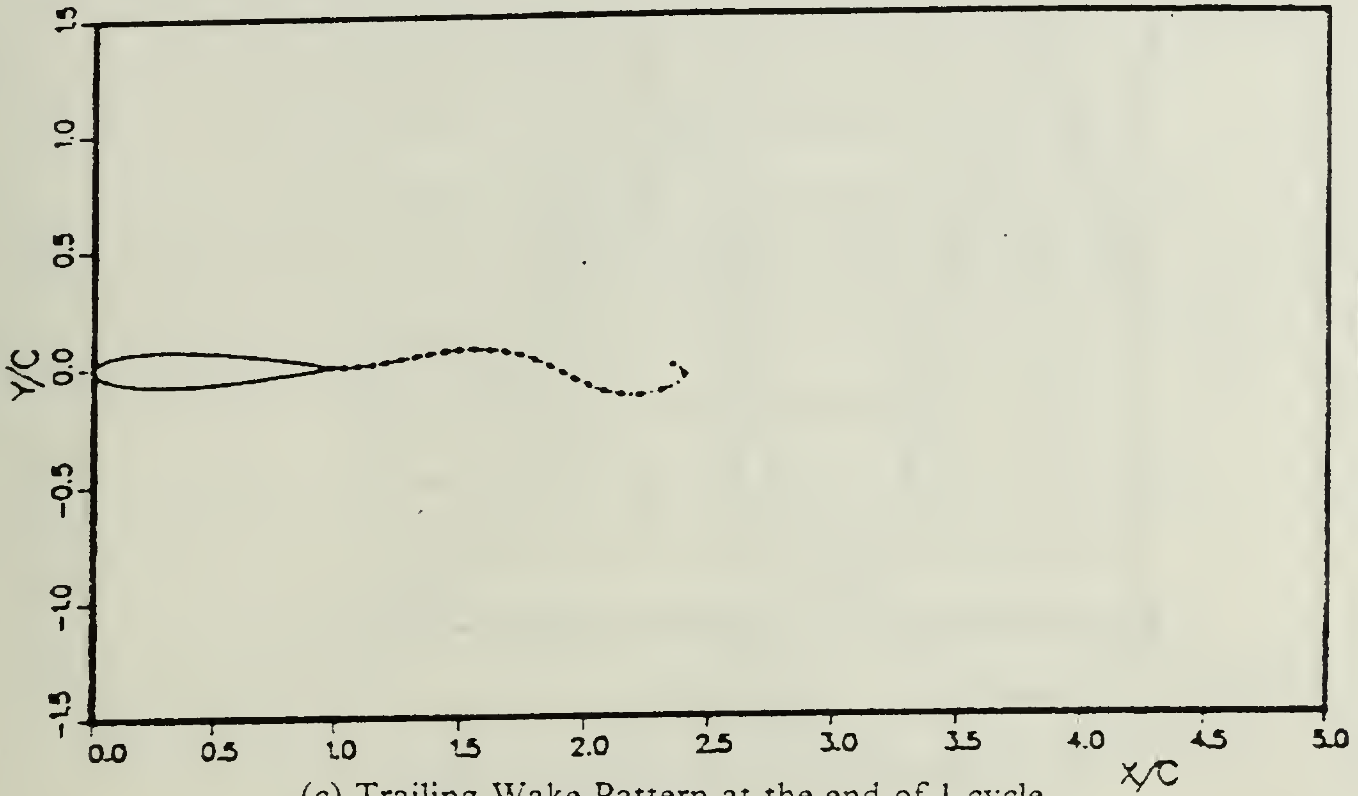
Figures 5.11 and 5.12 present the results of an airfoil executing a plunging motion at an amplitude of 0.018c but with two different reduced frequencies of 4.3 and 17.0 respectively. These numbers are chosen to coincide with those numbers used in [Ref. 4]. Excellent correlations are obtained. Notice from these Figures that the oscillation frequency has a great influence on the magnitudes of the aerodynamic parameters due to the formation of significantly different trailing wake patterns for the same oscillation amplitude. Also to note is that the width of the resulting trailing wake is much larger than the amplitude of the oscillation, reinforcing the fact that the unsteady flow is strongly governed by the shed vorticity in the trailing wake. The lift and pitching moment oscillate at the same frequency as the airfoil motion but slightly out of phase, the phase differences vary with the oscillation frequency. The drag is however oscillating at about twice the frequency of the airfoil motion with a negative mean value, indicating that the plunging action indeed generates some propulsive thrust. The same conclusion was arrived at in the experimental work of Halfman [Ref. 9] using a symmetrical NACA-0012 airfoil.



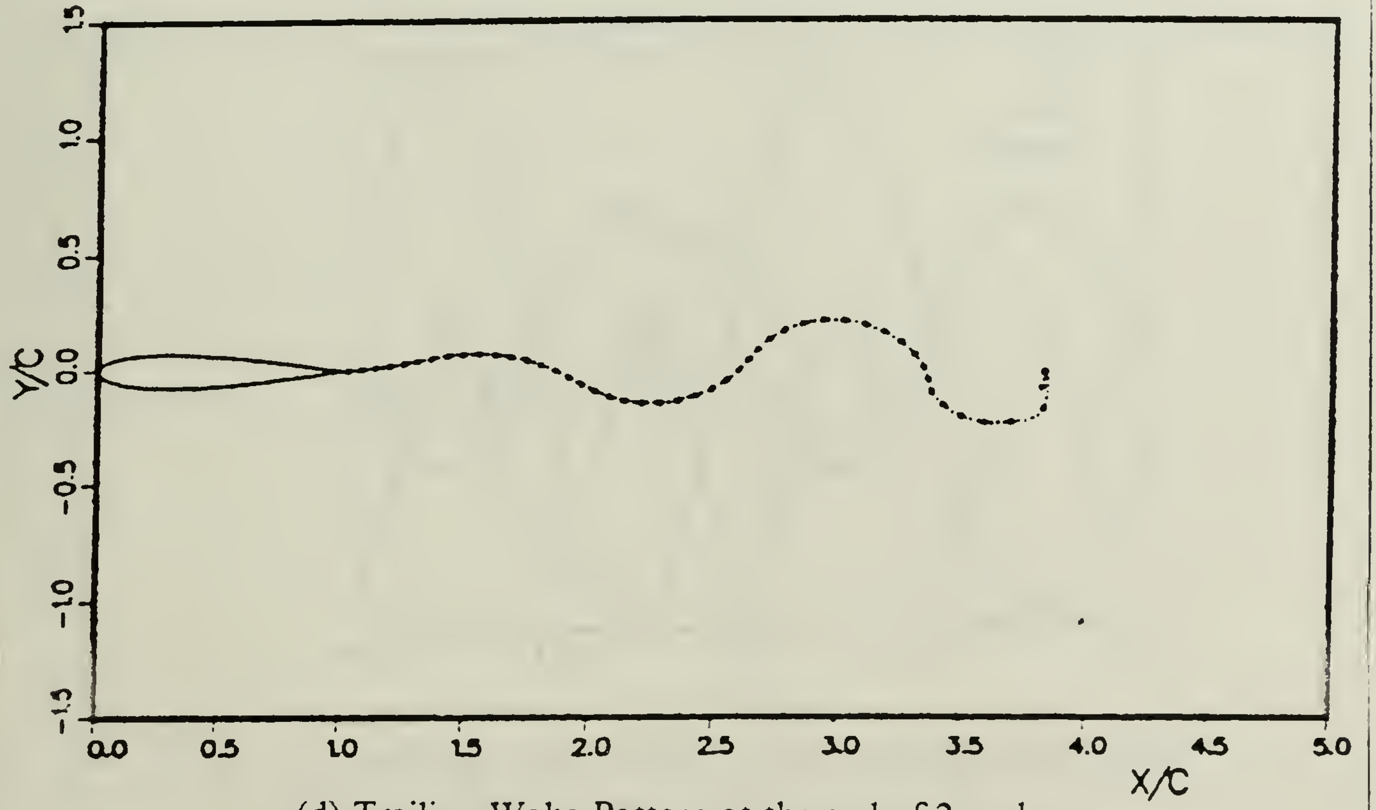
(a) Position of Airfoil (Positive Downward)

(b) Time History of C_ℓ, C_m and C_d over 2 Cycles.

Figure 5.11 Harmonic Plunging Motion of a NACA-0015 Airfoil
$\delta h_y = 0.018c$, $\omega c/V_\infty = 4.3$.

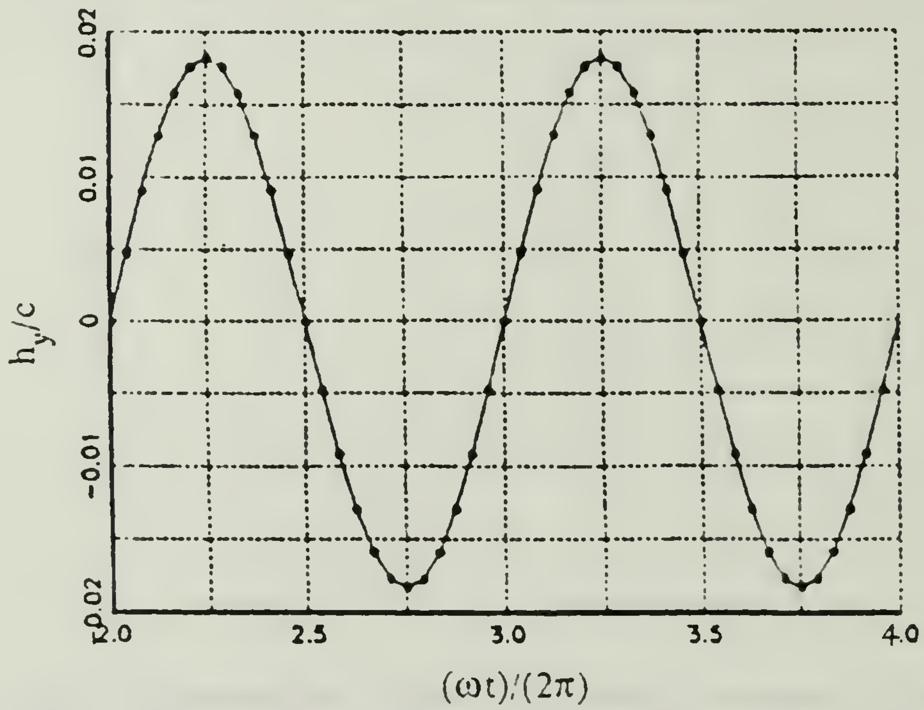


(c) Trailing Wake Pattern at the end of 1 cycle.

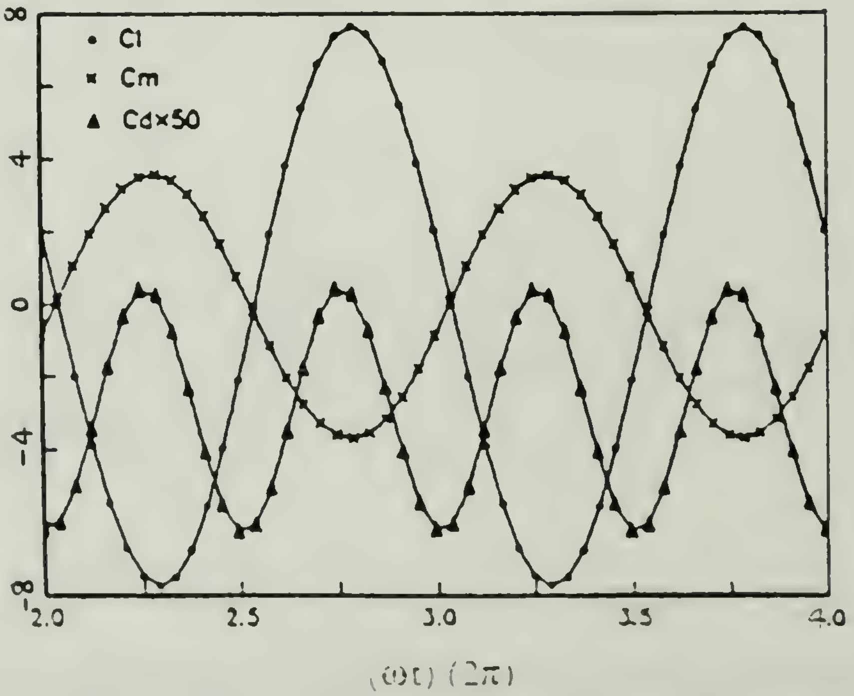


(d) Trailing Wake Pattern at the end of 2 cycles.

Figure 5.11 . (cont'd.)

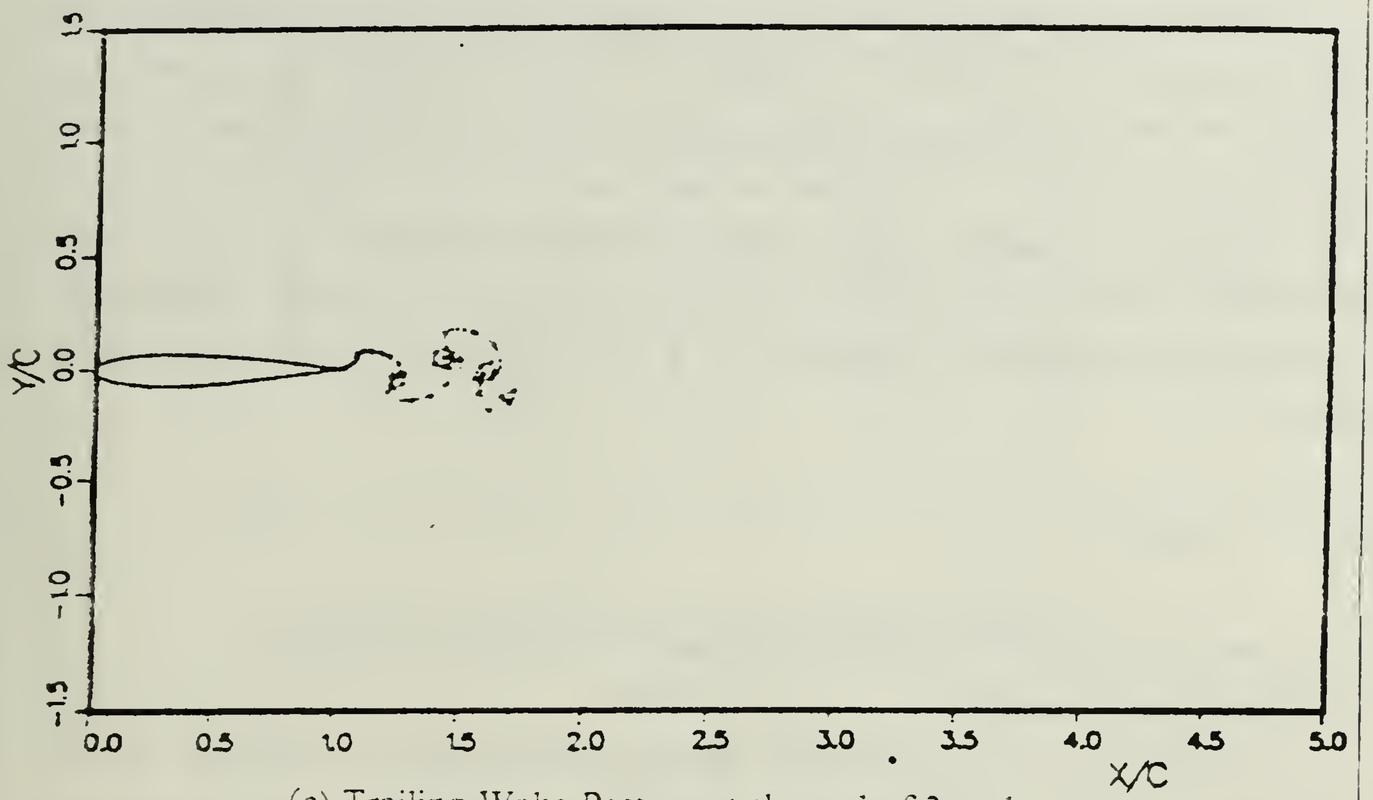


(a) Position of Airfoil (Positive Downward)

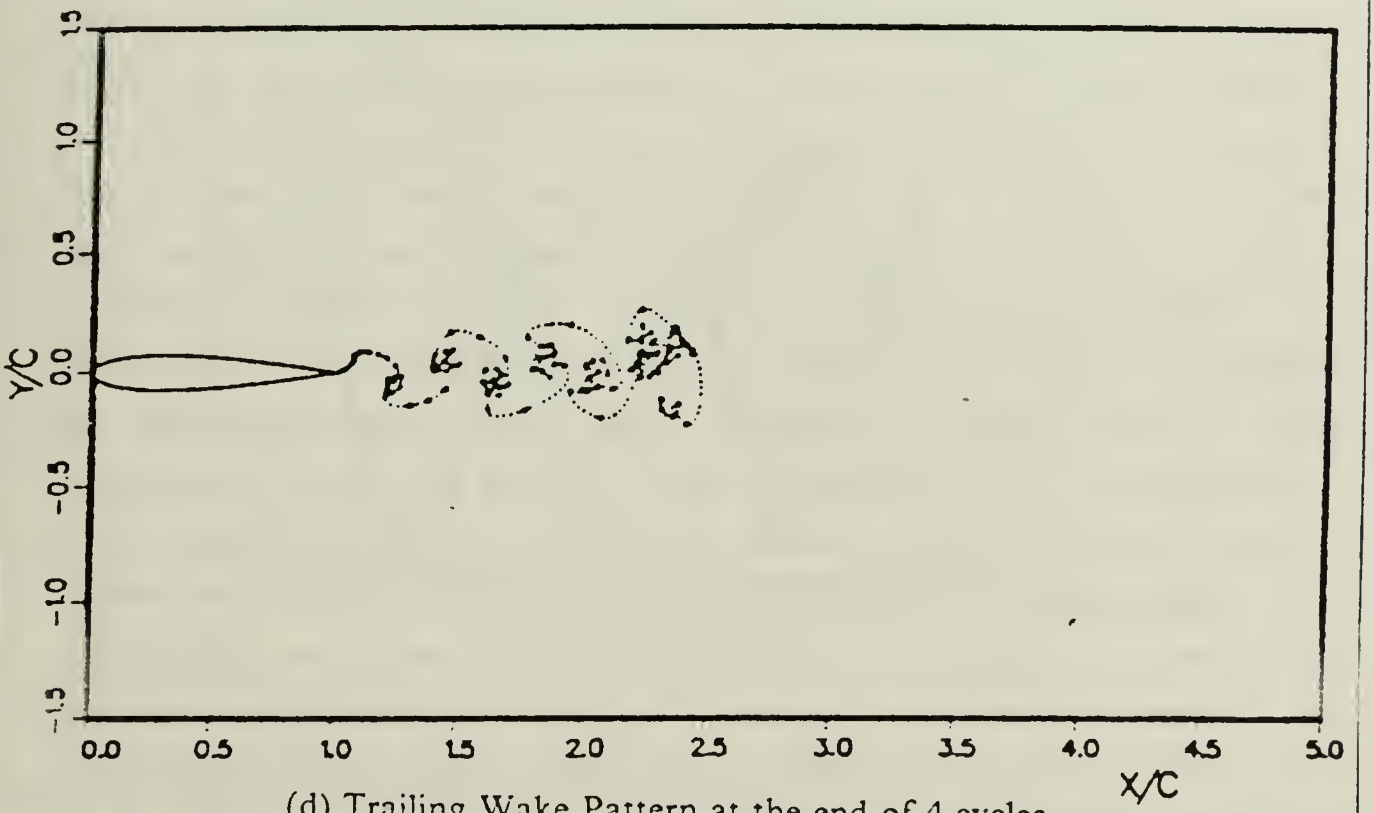


(b) Time History of C_ℓ, C_m and C_d' over 2 Cycles.

Figure 5.12 Harmonic Plunging Motion of a NACA-0015 Airfoil
$\bar{\delta} h_y = 0.018c$, $\omega c / V_\infty = 17.0$.



(c) Trailing Wake Pattern at the end of 2 cycles.



(d) Trailing Wake Pattern at the end of 4 cycles.

Figure 5.12 . (cont'd.)

D. ROTATIONAL HARMONIC OSCILLATION

1. Case-Run Definitions

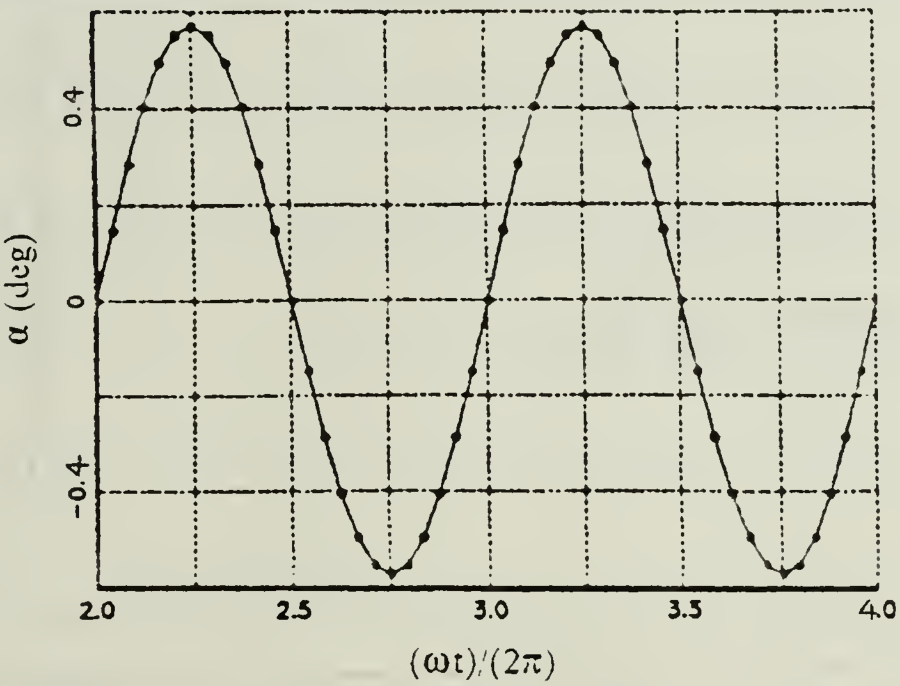
The case of an airfoil oscillating harmonically in pitch will be uniquely defined only if a pivot point is prescribed. If a fixed pivot point is used in the calculations, take for example the leading edge, any pitching motion about a pivot point other than the leading edge would need to be described as composed of a pitching and a translation motions about the leading edge. Program U2DIIF handles such conversion automatically without having the user to figure out the combined motion. This applies also to the case of modified ramp rise in AOA. The harmonic pitching oscillation is described by,

$$\alpha(t) = \delta\alpha \sin(\omega t)$$

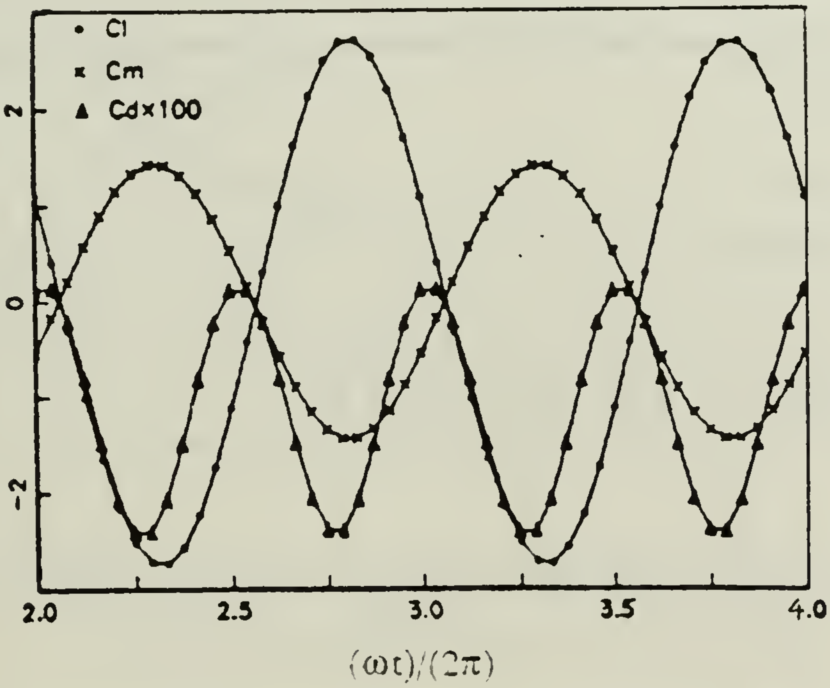
where $\delta\alpha$ and ω are the amplitude and frequency of oscillation respectively.

2. Results and Discussions

The results for the case of the 8.4% thick Von Mises symmetric airfoil, oscillating at an amplitude of 0.01 rad (or 0.573°) at a rather high reduced frequency of $\omega c/V_\infty = 20.0$ about the leading edge, are shown in Figure 5.13. The lift, drag and pitching moment time-history as well as the trailing wake patterns are very much similar to the case of a plunging airfoil at frequency of the same order of magnitude. The differences are clearly in the magnitudes and phase angles. These results check closely to those of [Ref. 3]. Figure 5.14 shows the results of the same Mises airfoil performing another harmonic oscillation at a lower reduced frequency of 0.8 and a large amplitude of 0.3973 rad about a pivot point 0.5 chord length ahead of the leading edge. [Ref. 4] conducted the same analysis for this pitch oscillation although the reason for using such a high amplitude of almost 23° was not clear. It is envisaged that such high amplitude may result in flow separation. Nevertheless, the case-run is carried out assuming validity of attached flow for the sake of comparing the results. Perhaps an inherent advantage, in the light of U2DIIF code verification, with the use of high amplitude in this case-run is that a discrepancy, if any, would show up significantly. Due to the use of different computation time-step size the pressure distributions on the airfoil, shown in Figure 5.14, do not correspond one-to-one at exactly the same angular positions as those presented in [Ref. 4]. However, the angular positions are matched to within 0.001°, 0.05° and 0.8° respectively for the three pressure distributions shown. They correlate very well to the results of [Ref. 4].

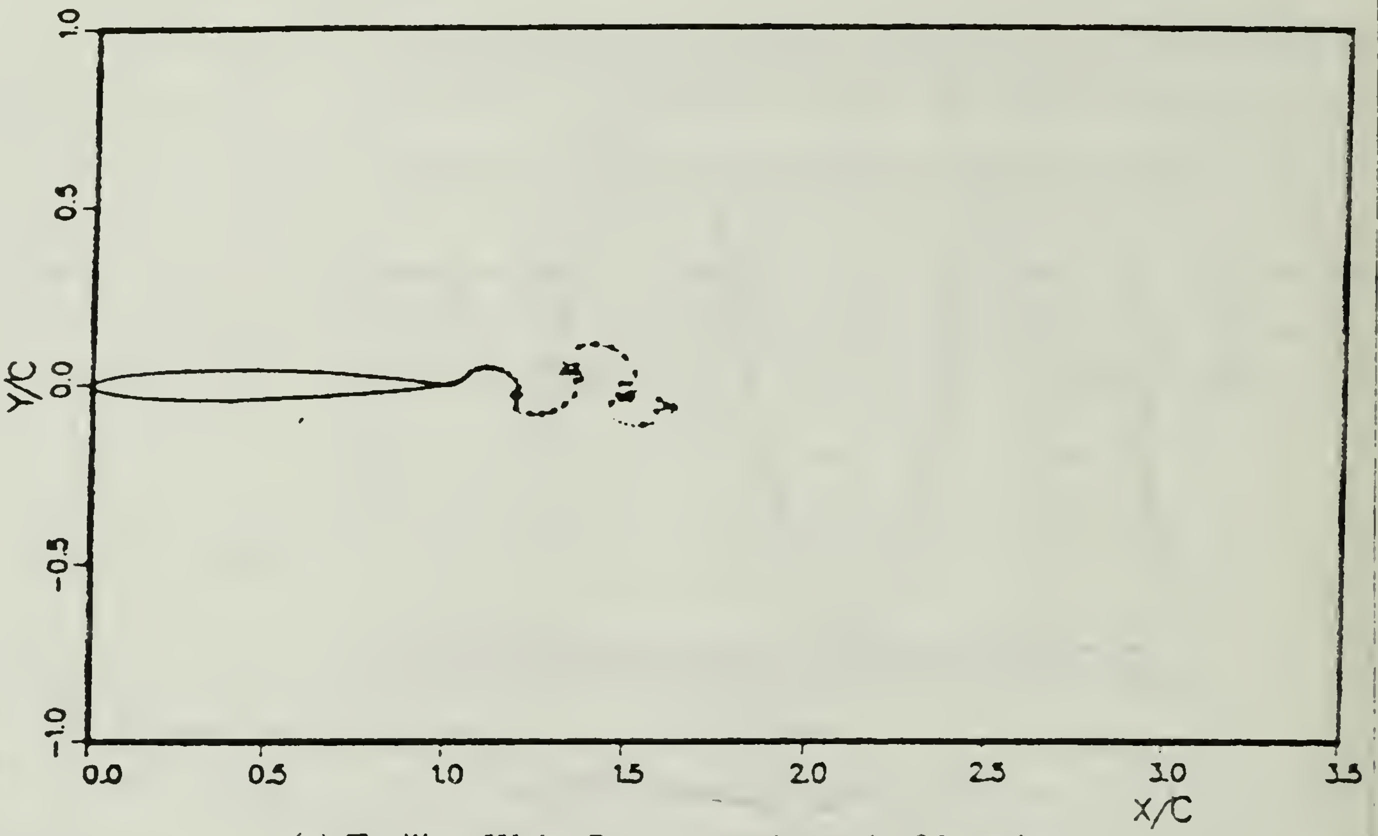


(a) Angular Position of Airfoil

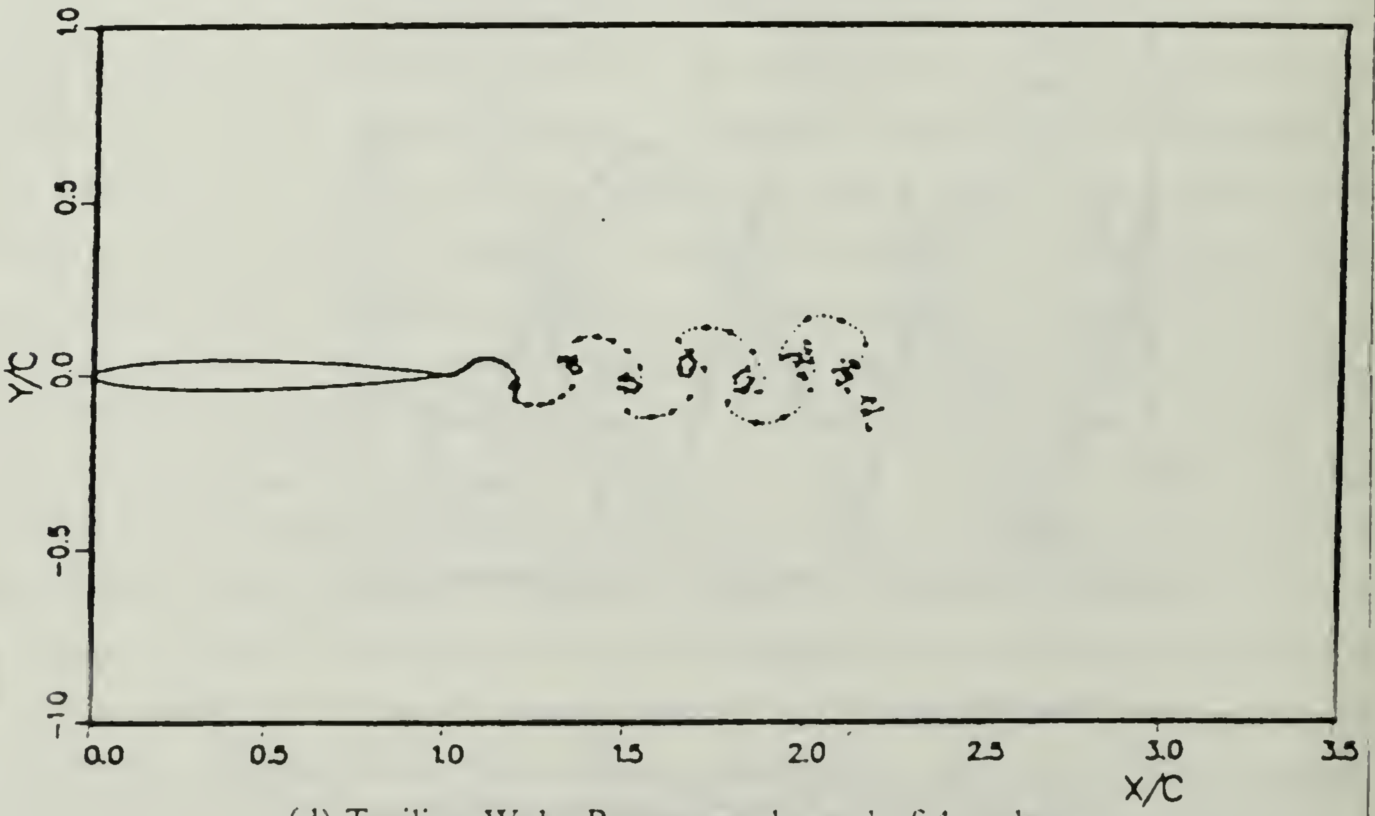


(b) Time History of C_ℓ, C_m and C_d over 2 Cycles.

Figure 5.13 Harmonic Pitching Motion about the Leading Edge of a 8.4% Thick Von Mises Airfoil
$\delta\alpha = 0.01$ rad, $\omega c/V_\infty = 20.0$.

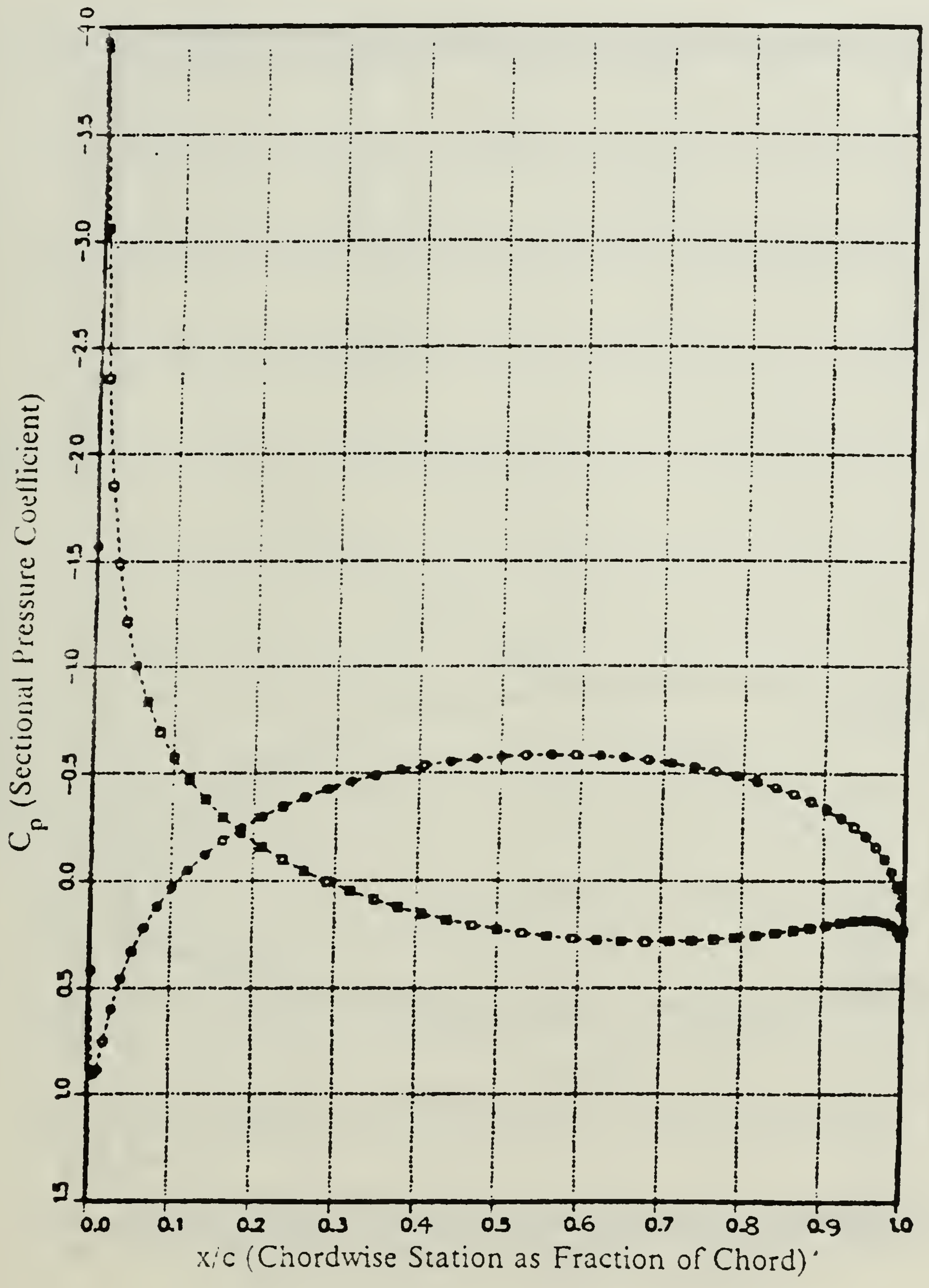


(c) Trailing Wake Pattern at the end of 2 cycles.



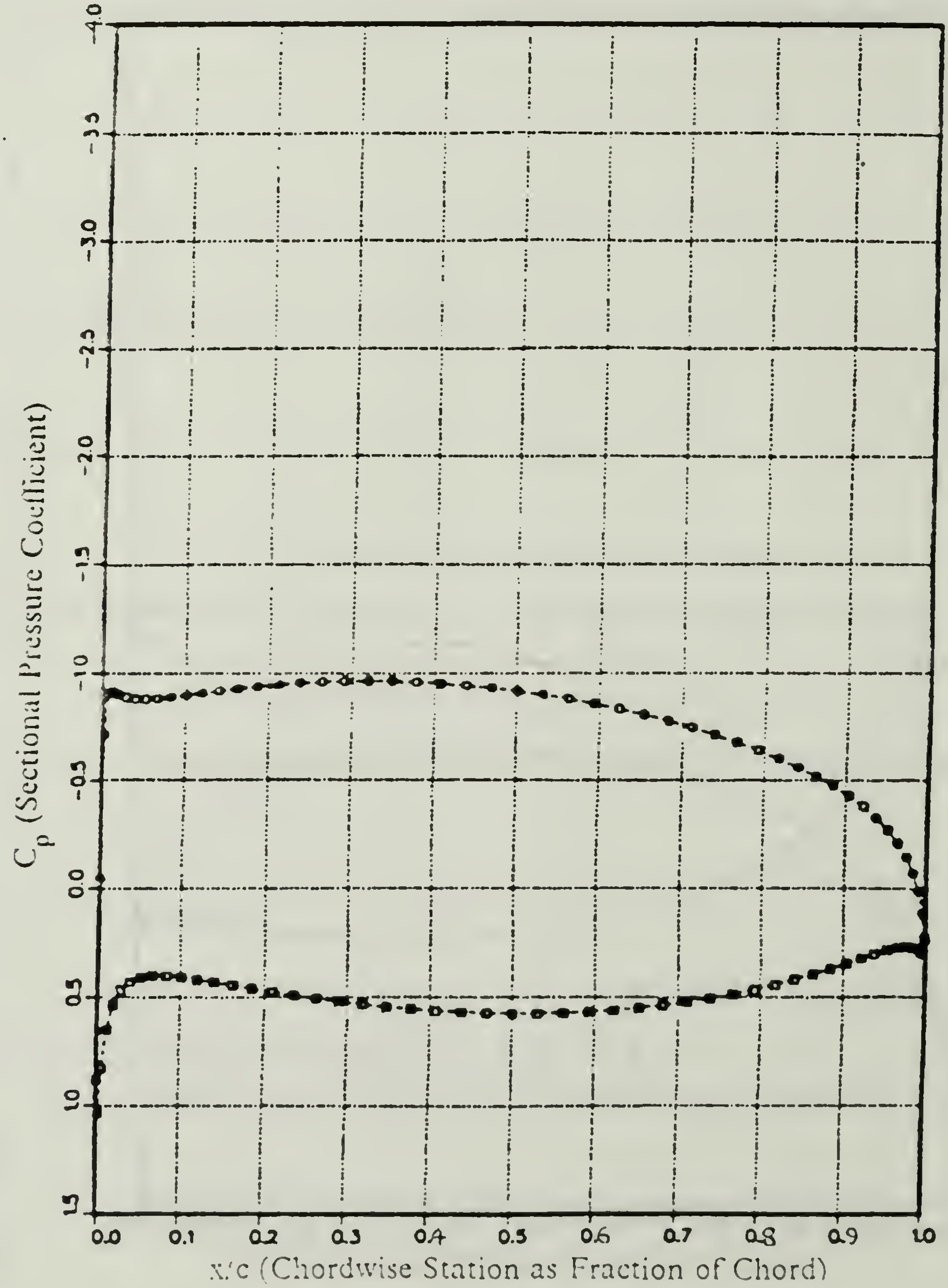
(d) Trailing Wake Pattern at the end of 4 cycles.

Figure 5.13 . (cont'd.)



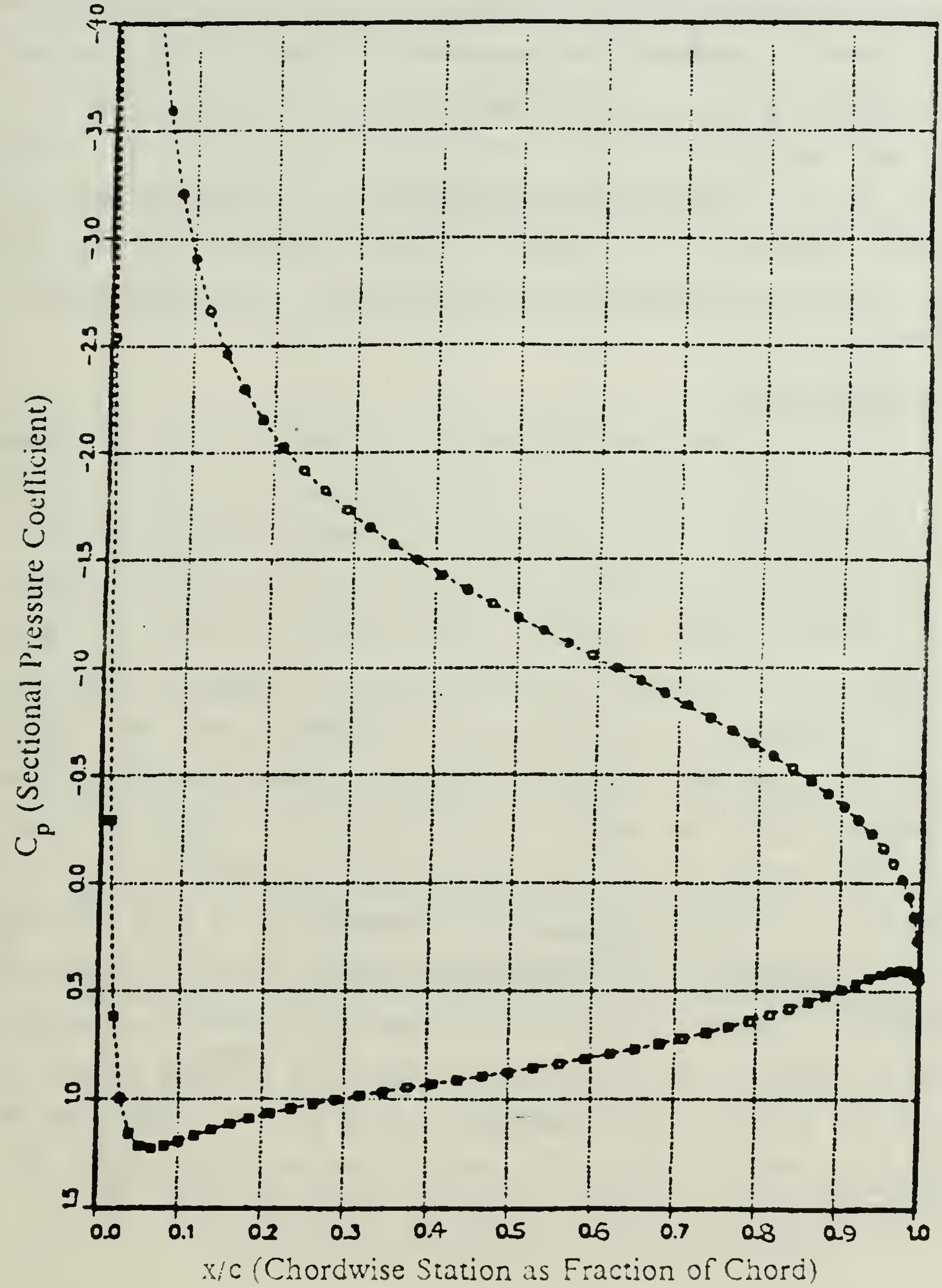
(a) Pressure Distribution at $\alpha = -0.2206$ rad.

Figure 5.14 Harmonic Pitching Motion about a Pivot 0.5c ahead of the Leading Edge of a 8.4% Thick Von Mises Airfoil
$\delta\alpha = 0.3973$ rad, $\omega c/V_\infty = 0.8$.



(b) Pressure Distribution at α = -0.0775 rad.

Figure 5.14 . (cont'd.)



(c) Pressure Distribution at α = 0.1876 rad.

Figure 5.14 . (cont'd.)

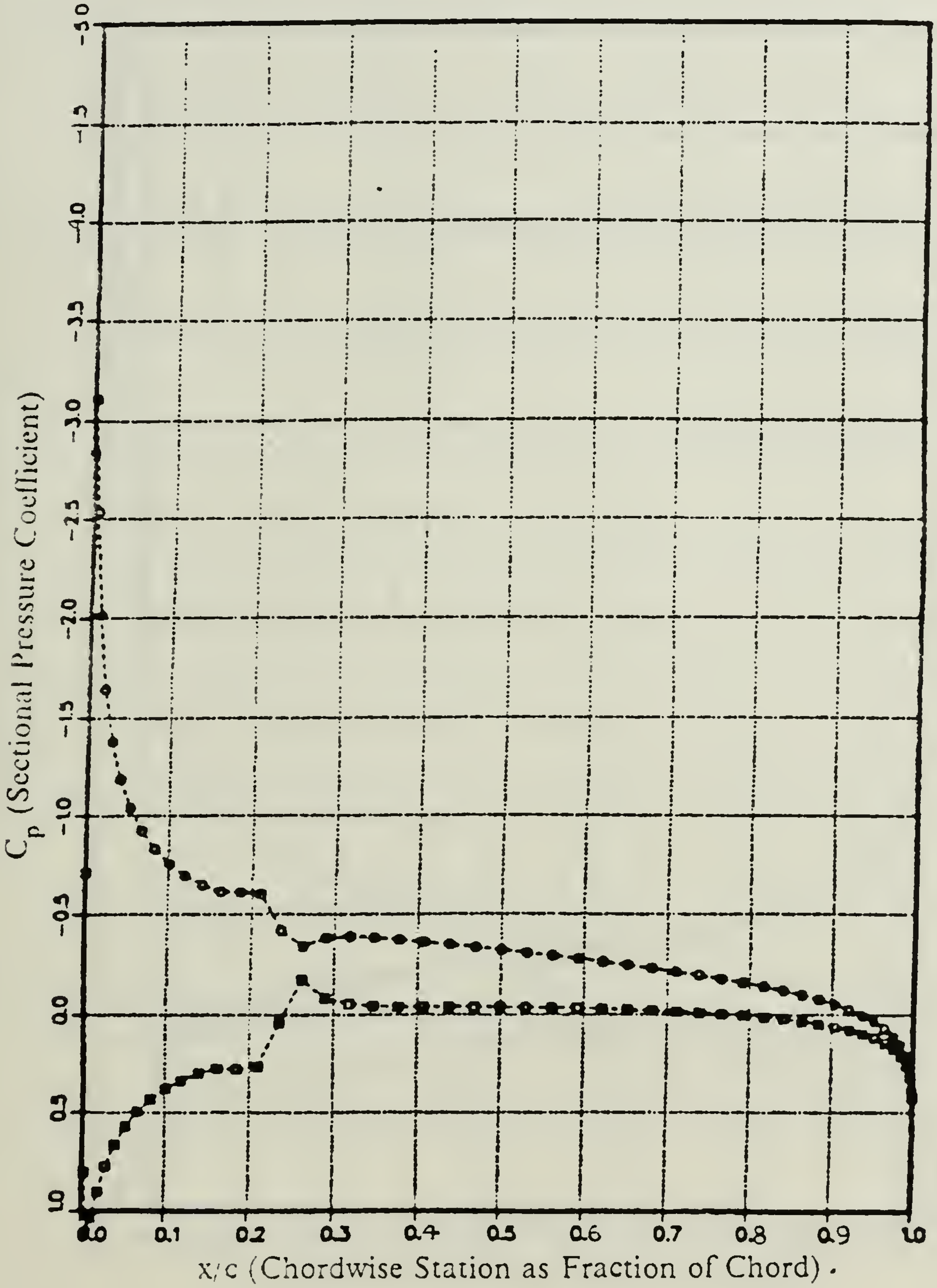
E. SHARP EDGE GUST FIELD PENETRATION

1. Case-Run Definitions

The case of an airfoil penetrating a sharp edge gust field can be computed using the U2DIIF code by specifying the components of gust velocity along and perpendicular to the freestream V_∞ and the angle of attack of the airfoil. The results that are presented here consider the case of airfoil penetrating a sharp edge vertical gust at zero AOA, in view of generating information on the time-dependent lift resembling the classical results of Küssner [Ref. 10] based on linearised theory. The gust front is taken to be at the leading edge at t_0 with only the transverse (vertical) component of $0.25V_\infty$.

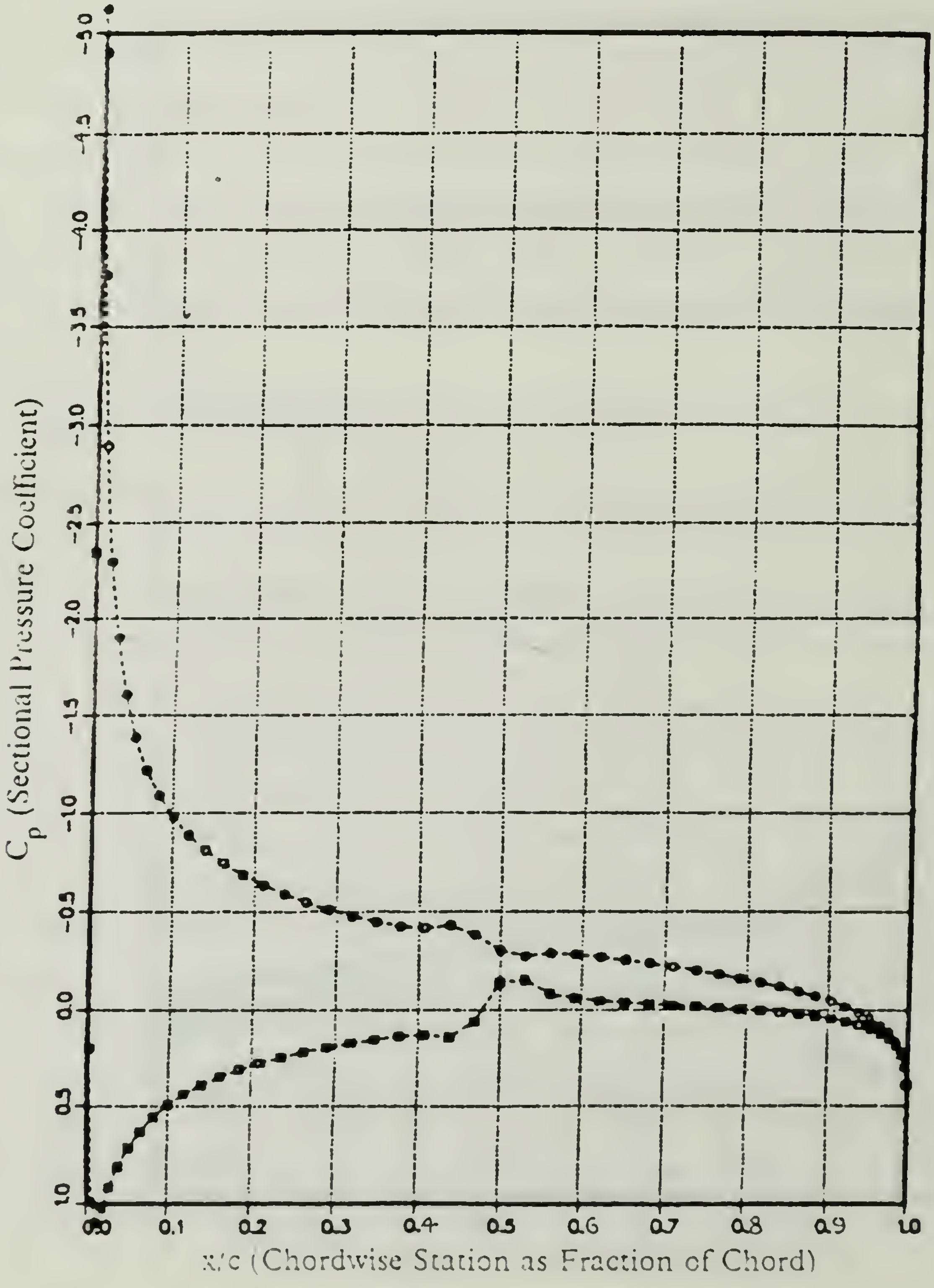
2. Results and Discussions

Figures 5.15 and 5.16 demonstrate the variation of pressure distributions and trailing wake patterns respectively during and shortly after the gust front moves past the airfoil. It is interesting to see that the resulting wake pattern after the entire airfoil is submerged in the gust field is as if being split by the gust front into two portions curling in opposite directions. [Ref. 3] predicted a similar behaviour for the case of an undeformed gust front. Due to the use of the modified flow tangency condition to handle the situation when the gust front happens to fall in between two nodal points, the pressure distributions, predicted by U2DIIF code, lie in between the results of the undeformed and deformed gust front models used in [Ref. 3]. We therefore conclude that this modified flow tangency condition produces sufficiently accurate results without adding complication in deformed gust field modeling which in turns limits the application to only sharp edge gusts. The present method would therefore preserve the generality for extending calculations to other types of gust front. Another comparison is made, as shown in Figure 5.17, by plotting the build-up of lift as a function of distance traveled by the airfoil in chord lengths. Shown in the same Figure are the Küssner Function and the results obtained from another case-run using a 25.5% thick Joukowski airfoil.



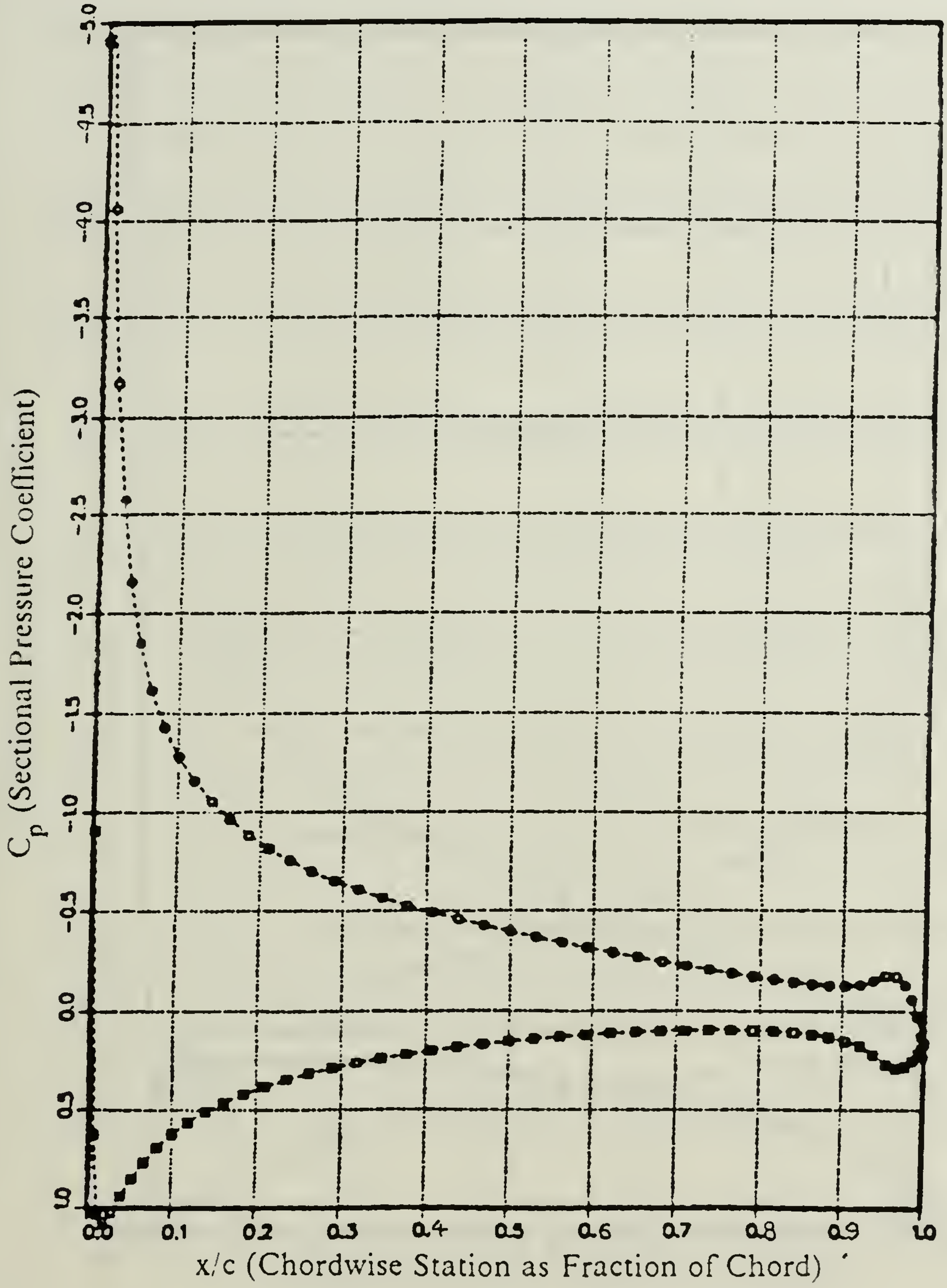
(a) $tV_{\infty}/c = 0.25$

Figure 5.15 Pressure Distributions at Various Time Instances Resulting from a 8.4% Thick Von Mises Airfoil Penetrating a Vertical Sharp Edge Gust of $0.25V_{\infty}$.



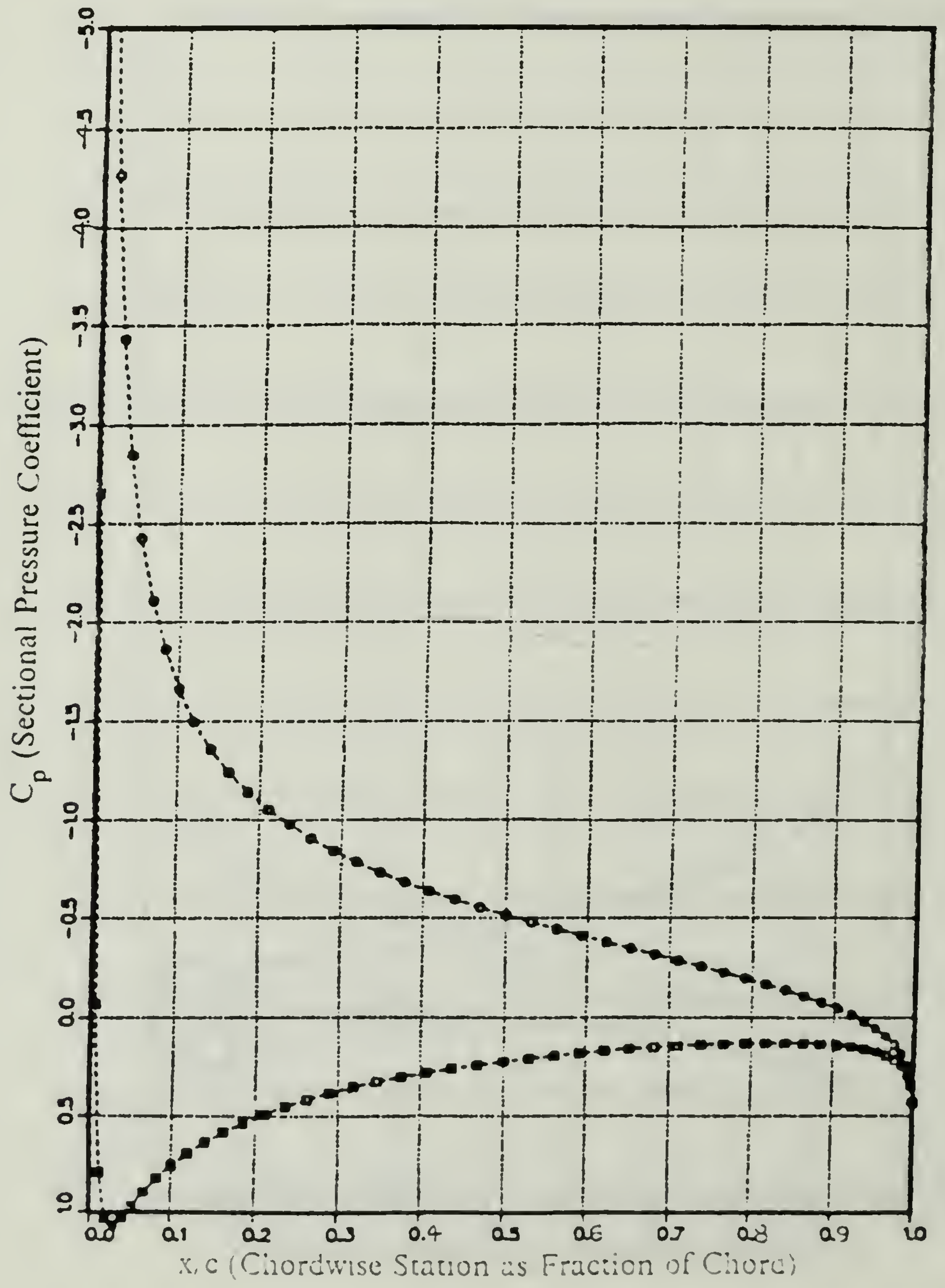
(b) $tV_{\infty}/c = 0.5$

Figure 5.15 . (cont'd.)



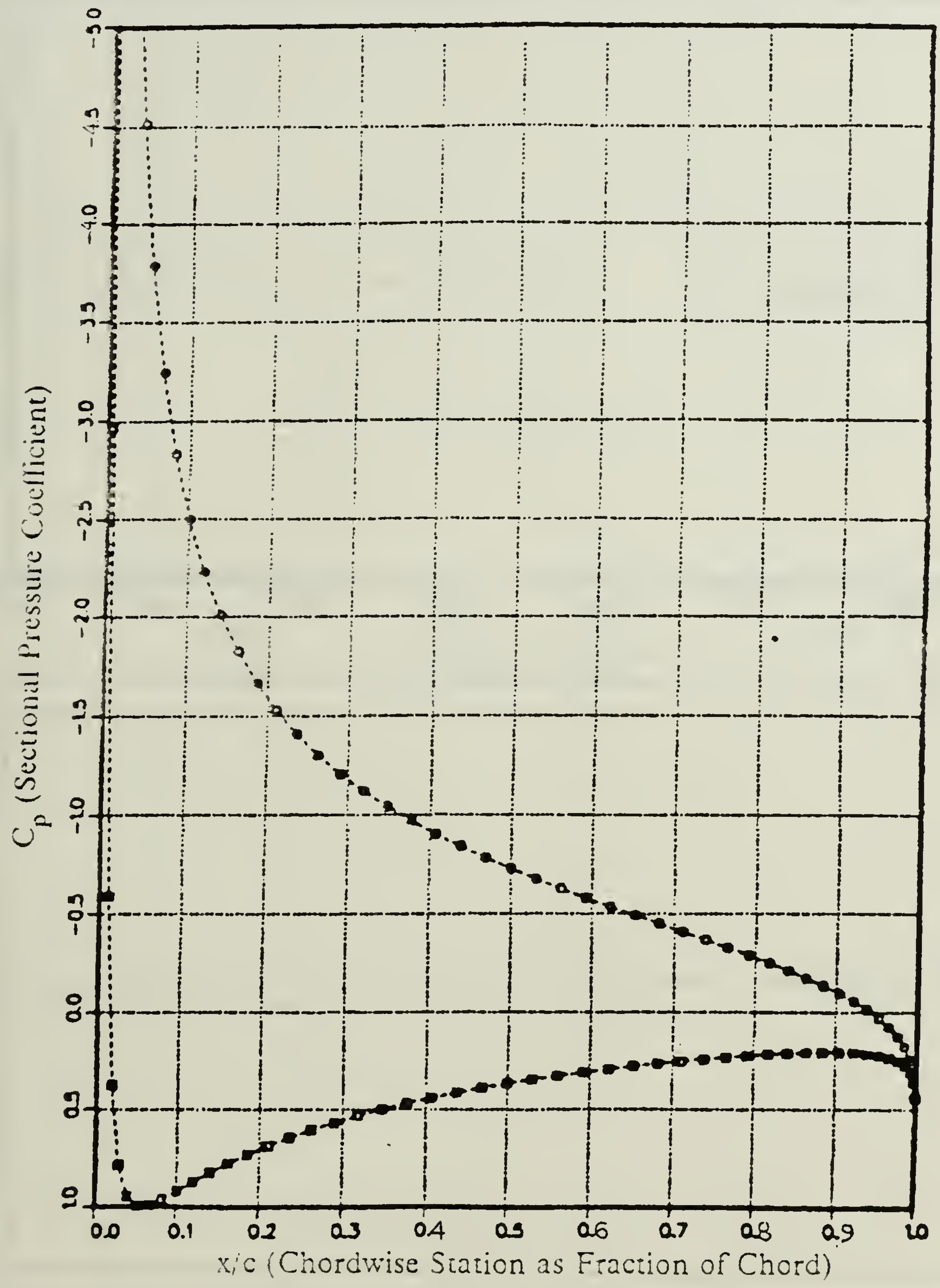
(c) $tV_\infty/c = 1.0$

Figure 5.15 . (cont'd.)



(d) $tV_{\infty}/c = 2.0$

Figure 5.15 . (cont'd.)



(e) $tV_{\infty}/c = \infty$

Figure 5.15 . (cont'd.)

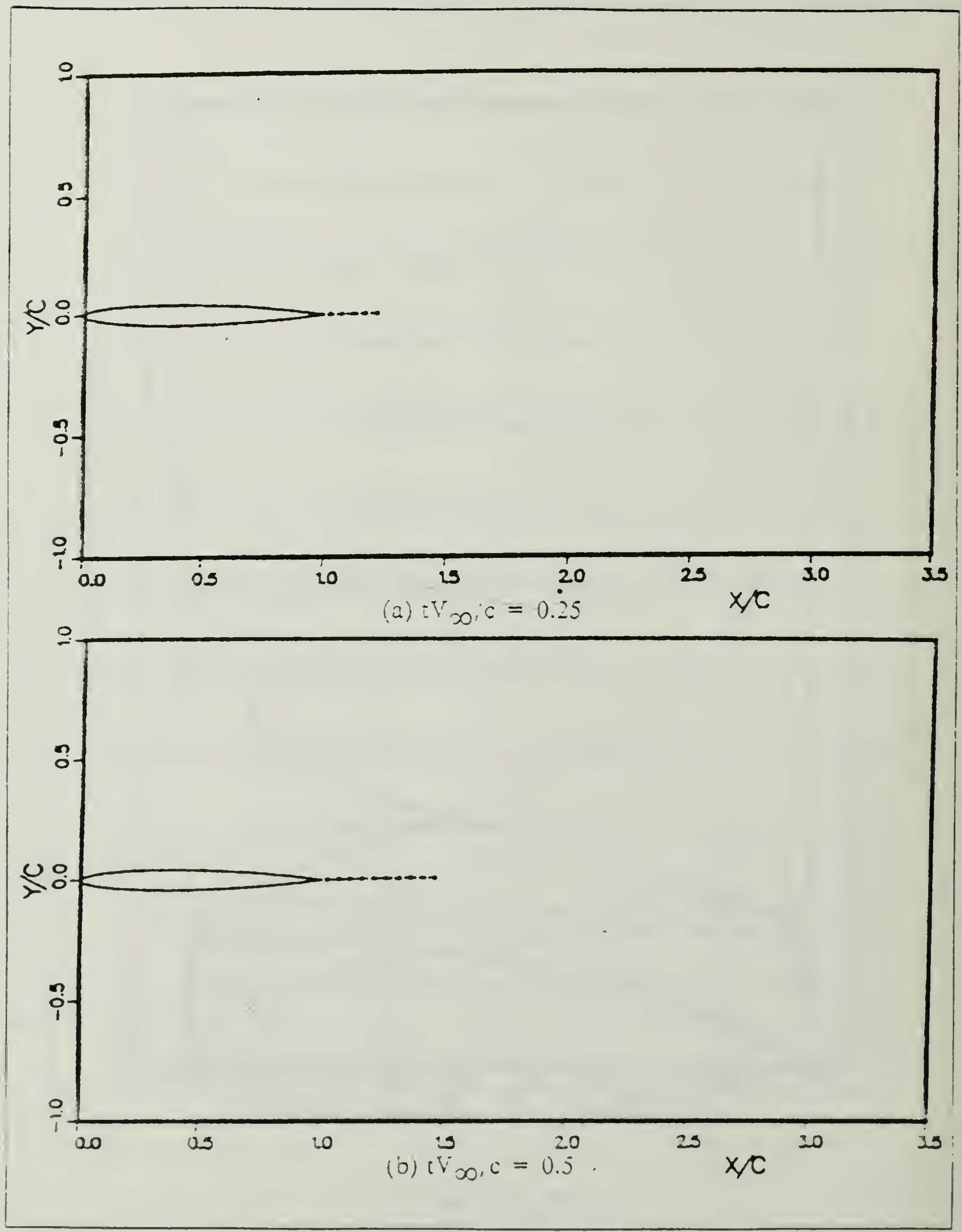
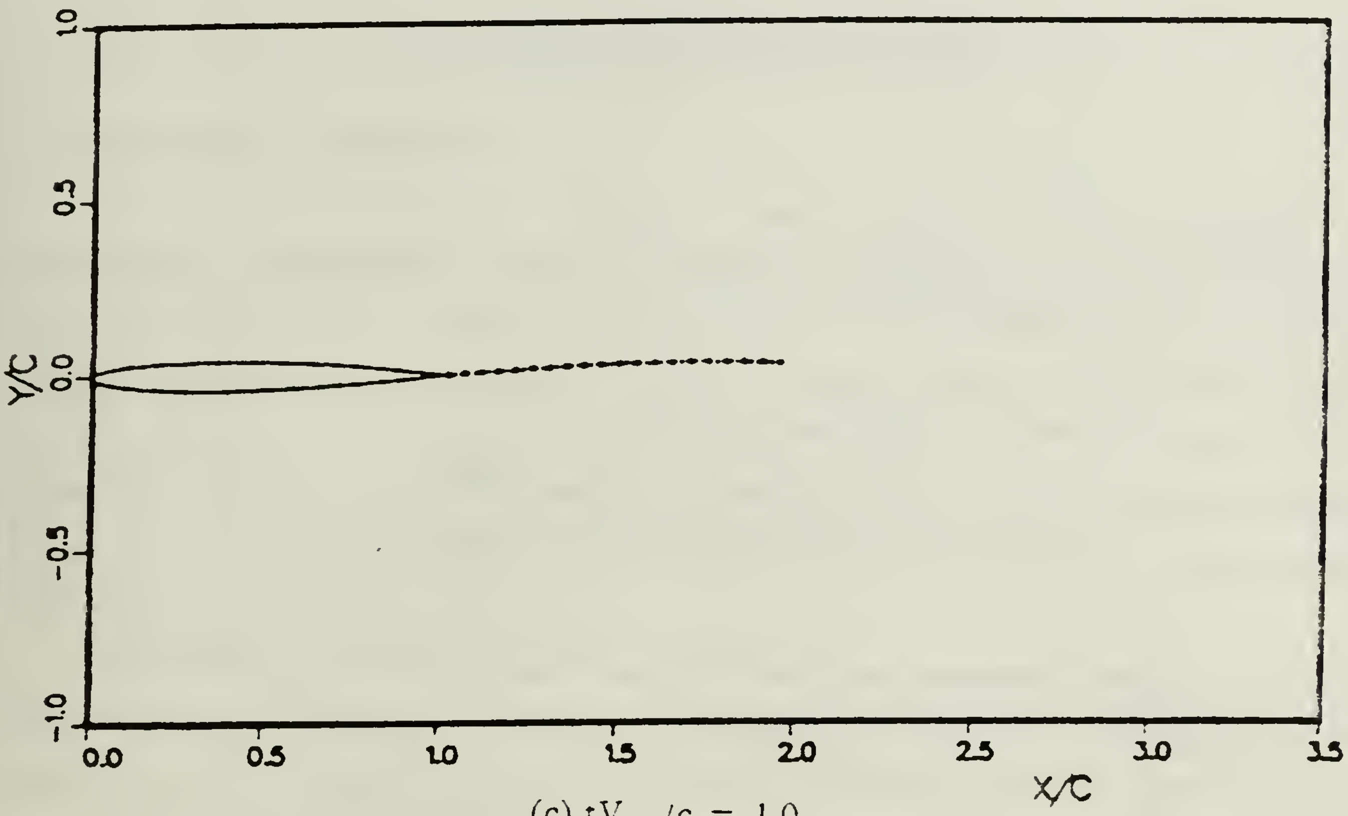
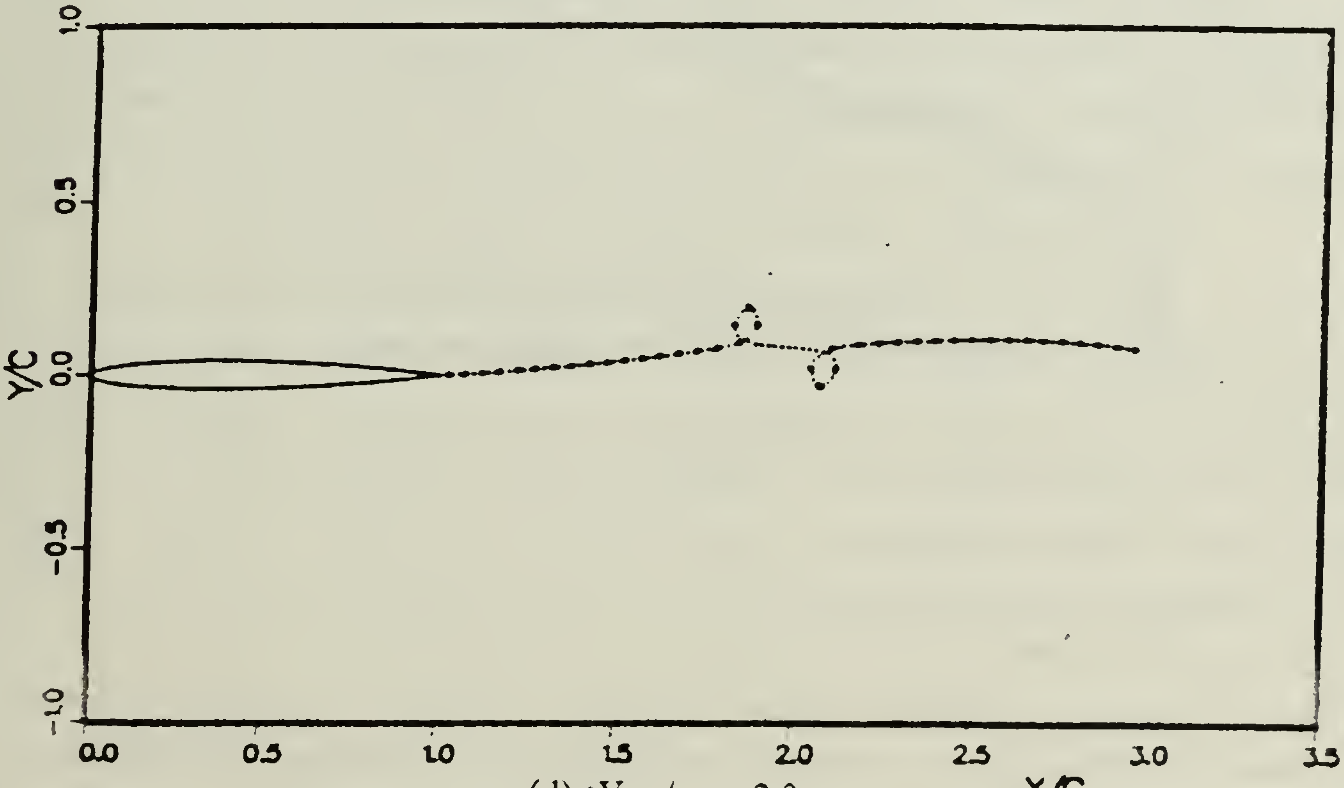


Figure 5.16 Trailing Wake Patterns at Various Time Instances Resulting from a 8.4% Thick Von Mises Airfoil Penetrating a Vertical Sharp Edge Gust of $0.25V_\infty$.



(c) $tV_\infty/c = 1.0$



(d) $tV_\infty/c = 2.0$

Figure 5.16 . (cont'd.)

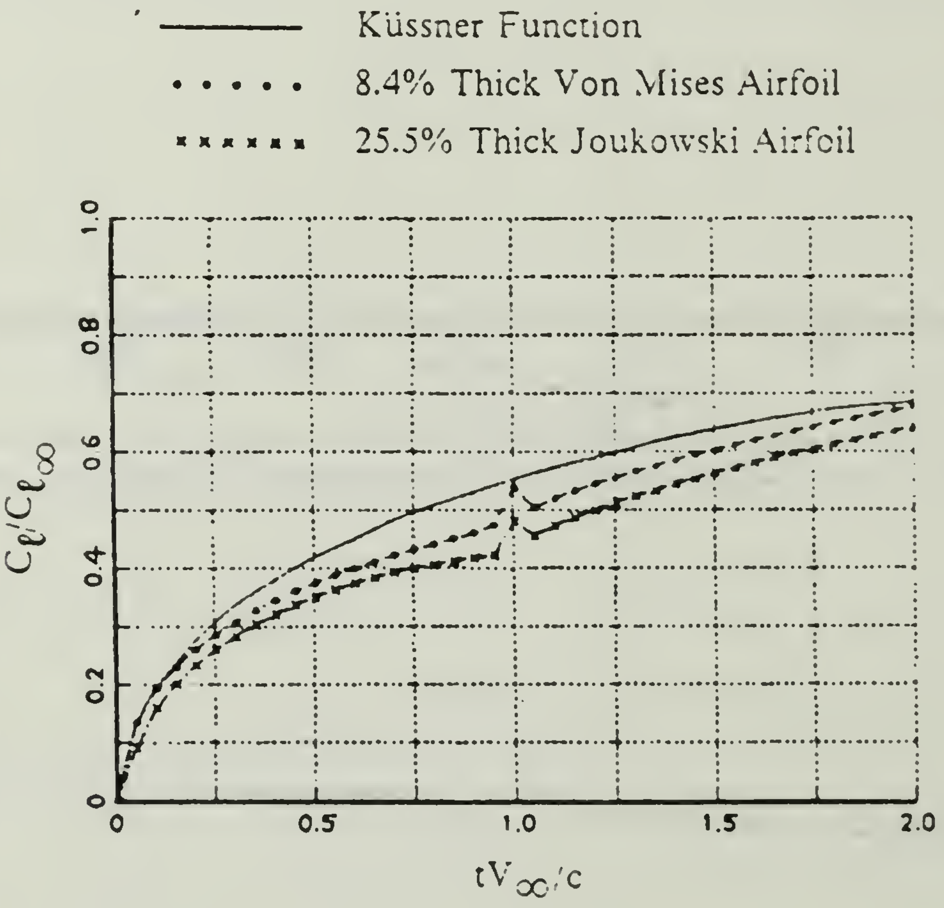


Figure 5.17 Normalised Time-Dependent Lift C_ℓ/C_{ℓ_∞} due to Airfoils of Various Thicknesses Penetrating a Vertical Sharp Edge Gust of $0.25V_\infty$.

VI. CONCLUDING REMARKS

A. GENERAL COMMENTS

The U2DIIF computer code has been developed for the purpose of demonstrating the successful extension of the well known panel methods, which have been used extensively for steady flow problems, into a powerful numerical tool for solving the unsteady flow problems. The mathematical modeling of the various types of unsteady flows has been done with the goal of preserving the generality of the methods. The intention is to present a method that has the minimum inherent limitations and restrictions so that its usage for future applications and developments could remain appealing.

The validation of the U2DIIF code has been done through the various case-runs of the numerous types of unsteady flow problems. The results of each case-run has been shown to be well correlated to the results obtainable from the literature in the form of theoretical analyses, numerical calculations based on different variants of panel singularities and in cases where limited experimental data are available. The ability of those case-runs using an airfoil as thin as 1% thickness to produce results that correlate accurately to the theoretical flat plate results is perhaps a remarkable robustness possessed by the present unsteady flow solution methods.

B. ENHANCING U2DIIF PROGRAM'S CAPABILITY

It has been noted in Chapter IV that the current U2DIIF code limits the total number of panels to 200 and the total computation time steps to 200 also. These limits are not at all rigid and can be easily increased if the computer storage space is not critical. A point to note is that the computation time will grow rapidly as these limits are raised. The current linear system solver, the Gaussian elimination algorithm, used in the code must be concurrently improved upon to efficiently reduce the computation time required for the iterations in each time step. A close examination of the matrix Equation 3.15, where the linear system solver is needed in every iteration, reveals that the coefficients of the left-hand-side matrix [A] are time-independent constants. Therefore the Gaussian elimination algorithm need only be done once for the entire unsteady flow calculations as far as the left-hand-side matrix is concerned. One could then perform, for each iteration, the manipulation of the two right-hand-sides

according to the steps taken for the reduction of the left-hand-side. This should cut down the computation time significantly against the current method of manipulating both the left- and right-hand-sides simultaneously in each iteration. The savings in computation time was not pursued in the development of U2DIIF code because the prime concern was to demonstrate that the basic iterative solution scheme works for the unsteady flow problems.

Another improvement that one may consider is to enable the code to be continuable from a time step where previous calculations were terminated. One sees this requirement necessary not only in the case of premature termination of computation due to some unforeseen circumstances, but also if one needs to prolong the computation time. Certainly with the current code structure, one has no choice but to perform the calculation right from the beginning.

A farther extension of U2DIIF code to the computation of unsteady flow problems involving multiple airfoils may be worth pursuing. Other research works that could be done based on U2DIIF code are in the area of incorporating more variety of rigid body motions into the code either in the form of closed-form equations or tabulated time history of the positions and rates of motions. It is important that one should use as close as possible, in the code, a rigid body motion that describes the physical motion before generating any numerical unsteady flow results for meaningful comparison to test data. This fact has been well illustrated and emphasized in the comparison of results of case-runs involving a step change to that of a ramp change in AOA with a fast rate which one could regard as a step in reality. However, remarkable difference in transient aerodynamics has been shown.

APPENDIX A
U2DIIF PROGRAM LISTINGS

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     PROGRAM U2DIIF                                                   C
C                                                                      C
C        UNSTEADY MOTION OF A TWO-DIMENSIONAL AIRFOIL                  C
C        IN INCOMPRESSIBLE INVISCID FLOW                               C
C        USING PANEL METHODS BASED ON THE HESS & SMITH                 C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /WAK2/ VYWK,VXWK
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CVVX(200),CVVY(200)
      COMMON /POT/ PHI(200),PHIK(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      COMMON /EXTV/ UE(200)
      DIMENSION XXC(200),YYC(200)
C
C        INPUT FROM FILE CODE 5 AND SET UP PANEL NODES AND SLOPES
C
      PI      = 3.1415926585
      WRITE (6,1003)
 1003 FORMAT (/////,' DATA READ FROM FILE CODE 1',//)
      CALL INDATA
      CALL SETUP
      READ (1,501) ALPI,DALP,TCON,FREQ,PIVOT,UGUST,VGUST
      WRITE (6,501) ALPI,DALP,TCON,FREQ,PIVOT,UGUST,VGUST
 501  FORMAT (7F10.6)
      READ (1,501) DELHX,DELHY,PHASE
      WRITE (6,501) DELHX,DELHY,PHASE
      READ (1,501) TF,DTS,TOL,TADJ
      WRITE (6,501) TF,DTS,TOL,TADJ
      IF (IFLAG .EQ. 0) WRITE (6,1005)
 1005 FORMAT (///,' COORDINATES OF AIRFOIL NODES',
     + //,3X,' X/C',6X,' Y/C',/)
      IF (IFLAG .EQ. 0) WRITE (6,1010) (X(I),Y(I),I=1,NODTOT+1)
 1010 FORMAT (F10.6,F10.6)
      WRITE (6,1020) SS
 1020 FORMAT(//,' AIRFOIL PERIMETER LENGTH = ',F10.6,/)
C
C     STEADY FLOW CALCULATION AT ALPI
C
      ALPHA   = ALPI
      WRITE (6,1030) ALPHA
 1030 FORMAT (//,' STEADY FLOW SOLUTION AT ALPHA = ',F10.6,/)
      IF (ALPHA .GT. 90.)   GO TO 200
      COSALF  = COS(ALPHA*PI/180.)
      SINALF  = SIN(ALPHA*PI/180.)
      CALL COFISH(SINALF,COSALF)
      CALL GAUSS(1)
      CALL VELDIS(SINALF,COSALF)
      CALL FANDM(SINALF,COSALF)
C
C        INITIALISATION FOR UNSTEADY FLOW CALCULATION TO BEGIN
C
      HX      = 0.0
      HY      = 0.0
      HXO     = 0.0
      HYO     = 0.0
```

```
      DHX     = 0.0
      DHY     = 0.0
      UX      = 0.0
      UY      = 0.0
      ALP = ALPI
      DA      = 0.0
      COSDA   = 1.0
      SINDA   = 0.0
      OMEGA   = 0.0
      XGF     = 0.0
      ANGLE   = ALPI*PI/180. + ATAN(VGUST/(1.+UGUST))
      COSANG  = COS(ANGLE)
      SINANG  = SIN(ANGLE)
      DO 100  IG = 1,NODTOT
      UG(IG)  = 0.0
 100  VG(IG)  = 0.0
      PHA     = PHASE*PI/180.
      VXW     = COSALF
      VYW     = SINALF
      GAMK    = GAMMA
      T       = 0.0
      M       = 0.0
      TOLD    = 0.0
C
C        RIGID BODY MOTIONS OF AIRFOIL
C
      IF (FREQ .NE. 0.0) GO TO 1
      IF (DALP .EQ. 0.0) GO TO 2
      IF (TCON .NE. 0.0) GO TO 3
      ALPHA   = ALPI + DALP
      COSALF  = COS(ALPHA*PI/180.)
      SINALF  = SIN(ALPHA*PI/180.)
 3    DT      = DTS
      TD      = DTS
      GO TO 60
 2    IF ((UGUST .EQ. 0.0) .AND. (VGUST .EQ. 0.0)) GO TO 200
      DT      = DTS
      TD      = DTS
      GO TO 60
 1    DT      = 2.0*PI/(FREQ*DTS)
      TD      = DT
 60   T       = DT
      WRITE (6,1051)
 1051 FORMAT (/////,' ****************************************',/,
     + ' ***  BEGIN UNSTEADY FLOW SOLUTION  ****',/,
     + ' ****************************************')
 40   M       = M + 1
      IF (T .GT. TF) GO TO 200
C
C        STORE CORE VORTEX COORDINATES FOR TIME STEP ADJUSTMENTS
C
      IF (M .EQ. 1) GO TO 50
      DO 51   I = 1,M-1
      XXC(I)  = XC(I)
 51   YYC(I)  = YC(I)
 50   IF (FREQ .NE. 0.0) GO TO 11
      IF (DALP .EQ. 0.0) GO TO 22
      IF (TCON .NE. 0.0) GO TO 33
C
C        STEP CHANGE IN AOA
C
      IF (TADJ .NE. 0.0) GO TO 70
      TD      = FLOAT(M+1)*DTS
      GO TO 70
C
C        MODIFIED RAMP CHANGE IN AOA
C
 33   IF (T .GT. TCON) GO TO 34
      DAL     = DALP * (3.-2.*T/TCON)*(T/TCON)**2
      ALPHA   = ALPI + DAL
```

```
      COSALF   = COS(ALPHA*PI/180.)
      SINALF   = SIN(ALPHA*PI/180.)
      DA       = ALPHA - ALP
      COSDA    = COS(DA*PI/180.)
      SINDA    = SIN(DA*PI/180.)
      OMEGA    = - (DALP*PI/180.) * (6.*T/(TCON*TCON)) * (1.-T/TCON)
      DHX      = PIVOT * (1.-COSDA)
      DHY      = - PIVOT * SINDA
      UY       = PIVOT * OMEGA
      MTCON    = M
      GO TO 70
 34   DAL      = 0.0
      ALPHA    = ALPI + DALP
      COSALF   = COS(ALPHA*PI/180.)
      SINALF   = SIN(ALPHA*PI/180.)
      DA       = 0.0
      COSDA    = 1.0
      SINDA    = 0.0
      OMEGA    = 0.0
      DHX      = 0.0
      DHY      = 0.0
      UY       = 0.0
      IF (TADJ .NE. 0.0) GO TO 70
      TD       = FLOAT(M+1-MTCON)*DTS
      GO TO 70
C
C        SHARP EDGE GUST (UGUST AND/OR VGUST)
C
 22   XGF      = T
      DO 110  IG = 1,NODTOT
      UG(IG)   = 0.0
      VG(IG)   = 0.0
      XG       = X(IG)*COSALF + Y(IG)*SINALF
      XGP1     = X(IG+1)*COSALF + Y(IG+1)*SINALF
      IF (IG .LT. NLOWER+1) GO TO 120
      IF (XGF .LE. XG) GO TO 110
      IF (XGF .GE. XGP1) GO TO 111
      FAC      = (XGF - XG)/(XGP1 - XG)
      UG(IG)   = UGUST*FAC
      VG(IG)   = VGUST*FAC
      GO TO 110
 111  UG(IG)   = UGUST
      VG(IG)   = VGUST
      GO TO 110
 120  IF (XGF .LE. XGP1) GO TO 110
      IF (XGF .GE. XG) GO TO 121
      FAC      = (XGF - XGP1)/(XG - XGP1)
      UG(IG)   = UGUST*FAC
      VG(IG)   = VGUST*FAC
      GO TO 110
 121  UG(IG)   = UGUST
      VG(IG)   = VGUST
 110  CONTINUE
      IF (XGF .LE. COSALF) MGUST = M
      IF (TADJ .NE. 0.0) GO TO 70
      IF (XGF .GT. COSALF) TD = FLOAT(M+1-MGUST)*DTS
      GO TO 70
C
C        TRANSLATION HARMONIC OSCILLATION
C
 11   IF (DALP .NE. 0.0) GO TO 12
      HX       = DELHX * SIN(FREQ*T + PHA)
      HY       = DELHY * SIN(FREQ*T)
      DHX      = HX - HXO
      DHY      = HY - HYO
      UX       = DELHX*FREQ*COS(FREQ*T+PHA)
      UY       = DELHY*FREQ*COS(FREQ*T)
      GO TO 70
C
C        ROTATIONAL HARMONIC OSCILLATION
```

```
C
 12     DAL      = DALP*SIN(FREQ*T)
         ALPHA   = ALPI + DAL
         COSALF  = COS(ALPHA*PI/180.)
         SINALF  = SIN(ALPHA*PI/180.)
         DA      = ALPHA - ALP
         COSDA   = COS(DA*PI/180.)
         SINDA   = SIN(DA*PI/180.)
         OMEGA   = - (DALP*PI/180.) * FREQ * COS(FREQ*T)
         UY      = PIVOT * OMEGA
         DHX     = PIVOT * (1.-COSDA)
         DHY     = - PIVOT * SINDA
C
C          TRANSFORM CORE VORTEX COORDINATES W. R. T. NEW AIRFOIL POSITION
C
 70     IF (M .EQ. 1) GO TO 80
         DO 90    I = 1,M-1
         XC(I)   = XXC(I) + CVVX(I) * DT
         YC(I)   = YYC(I) + CVVY(I) * DT
         XCO     = XC(I)
         YCO     = YC(I)
         XC(I)   = XCO*COSDA - YCO*SINDA + DHX
 90     YC(I)    = XCO*SINDA + YCO*COSDA + DHY
 80     CONTINUE
         WRITE (6,1001) T,DT
 1001 FORMAT (/////,' TIME STEP TK = ',F10.6,10X,'TK - TKM1 = ',F10.6,/)
         WRITE (6,1004) ALPHA,OMEGA,UX,UY
 1004 FORMAT (/,' ALPHA(T) = ',F10.6,5X,' OMEGA(T) = ',F10.6,/,
      +           ' U(T)      = ',F10.6,5X,' V(T)      = ',F10.6,///,
      + 1X,' NITR     VXW        VYW       WAKE       THETA      GAMMA',/)
C
C       CALCULATE THE TRAILING EDGE WAKE ELEMENT
C
         NITR    = 0
 10     WAKE     = SQRT(VYW*VYW+VXW*VXW)*DT
         THENP1  = ATAN2(VYW,VXW)
         COSTHE(NP1) = COS(THENP1)
         SINTHE(NP1) = SIN(THENP1)
         WRITE (6,1002) NITR,VXW,VYW,WAKE,THENP1,GAMK
 1002 FORMAT (I5,4F10.6,E14.6)
         X(NP2)  = X(NP1) + WAKE*COSTHE(NP1)
         Y(NP2)  = Y(NP1) + WAKE*SINTHE(NP1)
         CALL INFL (NITR)
         CALL COEF (SINALF,COSALF,OMEGA,UX,UY)
         CALL GAUSS(2)
         CALL KUTTA (ALPHA,SINALF,COSALF,OMEGA,UX,UY)
         CALL TEWAK (SINALF,COSALF)
         TOL1    = ABS(VYW - VYWK)
         TOL2    = ABS(VXW - VXWK)
         IF ((TOL1 .LT. TOL) .AND. (TOL2 .LT. TOL)) GO TO 20
         VYW     = VYWK
         VXW     = VXWK
         IF (NITR .GT. 200) STOP
         NITR    = NITR + 1
         GO TO 10
 20     WRITE (6,1011) NITR
 1011 FORMAT (//,' CONVERGED SOLUTION OBTAINED AFTER NITR = ',I3)
         CALL PRESS (SINALF,COSALF,OMEGA,UX,UY)
         IF ((UGUST .EQ. 0.0) .AND. (VGUST .EQ. 0.0)) GO TO 300
         CALL FANDM (SINANG,COSANG)
         GO TO 400
 300    CALL FANDM (SINALF,COSALF)
 400    CONTINUE
C
C       ADJUST TIME STEP (TADJ .NE. 0.0) IF NECESSARY
C
         IF (TADJ .EQ. 0.0) GO TO 95
         WRITE (5,2001)
 2001 FORMAT (//,' DO YOU WANT TO ADJUST TIME STEP ? 0 - NO, 1 - YES')
         READ (5,*) IDT
```

```
      IF (IDT .EQ. 0) GO TO 95
      DT       = TADJ * DT
      T        = TOLD + DT
      WRITE (6,1006)
 1006 FORMAT (//,' BACK-TRACK COMPUTATION AND ADJUST TIME-STEP',//)
      GO TO 50
C
C       WAKE ELEMENT LEAVES TRAILING EDGE AS A CORE-VORTEX
C
 95   CV(M)    = SS*(GAMMA-GAMK)
      XC(M)    = X(NP1) + 0.5*WAKE*COSTHE(NP1)
      YC(M)    = Y(NP1) + 0.5*WAKE*SINTHE(NP1)
      CVVX(M) = VXW
      CVVY(M) = VYW
      WRITE (6,1052)
 1052 FORMAT (//,' TRAILING VORTICES DATA',//,
     + 4X,'M',4X,'X(M)',6X,'Y(M)',6X,'CIRC',/)
      DO 900  I = 1,M
 900  WRITE (6,1050) I,XC(I),YC(I),CV(I)
 1050 FORMAT(I5,3F10.6)
      CALL CORVOR (SINALF,COSALF)
C
C       RE-INITIALISE PARAMETERS FOR NEXT TIME STEP CALCULATION
C
      DO 30    I = 1,NODTOT
      Q(I)     = QK(I)
      PHI(I)   = PHIK(I)
 30   CONTINUE
      GAMMA    = GAMK
      ALP      = ALPHA
      HXO      = HX
      HYO      = HY
      TOLD     = T
      DT       = TD
      T        = T + TD
      GO TO 40
 200  STOP
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE BODY(Z,SIGN,X,Y)                                      C
C                                                                      C
C             RETURN COORDINATES OF POINT ON THE BODY SURFACE          C
C                                                                      C
C                  Z = NODE-SPACING PARAMETER                          C
C                  X,Y = CARTESIAN COORDINATES                         C
C                  SIGN = +1. FOR UPPER SURFACE                        C
C                         -1. FOR LOWER SURFACE                        C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE BODY(Z,SIGN,X,Y)
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
      IF (SIGN .LT. 0.0)    Z = 1. - Z
      CALL NACA45(Z,THICK,CAMBER,BETA)
      X        = Z - SIGN*THICK*SIN(BETA)
      Y        = CAMBER + SIGN*THICK*COS(BETA)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE COEF (SINALF,COSALF,OMEGA,UX,UY)                      C
C                                                                      C
C             SET COEFFICIENTS OF N EQUS ARISING FROM FLOW             C
C                  TANGENCY CONDITIONS AT MID POINTS OF PANELS         C
C             SOLVING THE N-SOURCE STRENGTHS IN TERMS OF THE           C
C                  VORTICITY STRENGTH (RESULTING IN 2 RHS)             C
C             KUTTA CONDITION IS SATISFIED SEPARATELY TO OBTAIN        C
C                  THE VORTICITY STRENGTH                              C
C             THIS SOLUTION METHOD IS DESIRED FOR UNSTEADY FLOW        C
C                                                                      C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE COEF (SINALF,COSALF,OMEGA,UX,UY)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /COF/ A(201,211),NEQS
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CCVX(200),CCVY(200)
      COMMON /INF1/ AAN(201,201),BBN(201,201),AYNP1(201),BYNP1(201)
      COMMON /INF2/ CCN(201,200),CCT(201,200),CYNP1(200),CXNP1(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      NEQS    = NODTOT
      NP1     = NODTOT + 1
      NP2     = NODTOT + 2
C
C           INITIALISE COEFFICIENTS
C
      DO 90   I = 1,NODTOT
      DO 90   J = 1,NP2
 90   A(I,J)  = 0.0
C
C           SET LHS MATRIX A(I,J)
C
      DO 120  I = 1,NODTOT
      XMID    = 0.5 * (X(I) + X(I+1))
      YMID    = 0.5 * (Y(I) + Y(I+1))
      B       = 0.0
      DO 110  J = 1,NODTOT
      A(I,J)  = AAN(I,J)
      B       = B + BBN(I,J)
 110  CONTINUE
C
C           FILL IN THE RIGHT HAND SIDE
C
      A(I,NP1) = -B + BBN(I,NP1)*SS/WAKE
      A(I,NP2) = -BBN(I,NP1)*GAMMA*SS/WAKE
     + + SINTHE(I)* ((1.+UG(I))*COSALF-VG(I)*SINALF+UX)
     + - COSTHE(I)* ((1.+UG(I))*SINALF+VG(I)*COSALF+UY)
     +              + OMEGA*(YMID*SINTHE(I) + XMID*COSTHE(I))
C
C       ADD CORE VORTEX CONTRIBUTION
C
      IF (M .EQ. 1) GO TO 140
      MM1     = M - 1
      DO 100  N = 1,MM1
      A(I,NP2) = A(I,NP2) - CCN(I,N)*CV(N)
 100  CONTINUE
 140  CONTINUE
 120  CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE COFISH(SINALF,COSALF)                                 C
C                                                                      C
C           SET COEFFICIENTS OF LINEAR SYSTEM - N+1 EQUATIONS          C
C                N EQUS - FLOW TANGENCY AT MID POINTS OF PANELS        C
C                1 EQU  - KATTA CONDITION AT TRAILING EDGE PANELS      C
C           THIS SOLUTION METHOD IS EFFECTIVE FOR STEADY FLOW, NO      C
C                ITERATION IS REQUIRED, N-SOURCE STRENGTHS AND 1       C
C                VORTICITY STRENGTH ARE SOLVED SIMULTANEOUSLY          C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE COFISH(SINALF,COSALF)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /COF/ A(201,211),KUTTA
      COMMON /NUM/ PI,PI2INV
      KUTTA   = NODTOT + 1
C
```

```
C           INITIALISE COEFFICIENTS
C
      DO 90   J = 1,KUTTA
 90   A(KUTTA,J)    = 0.0
C
C           SET VN = 0 AT MID-POINT OF I-TH PANEL
C
      DO 120  I = 1,NODTOT
      XMID    = .5*(X(I) + X(I+1))
      YMID    = .5*(Y(I) + Y(I+1))
      A(I,KUTTA) = 0.0
C
C           --   FIND CONTRIBUTION OF J-TH PANEL
C
      DO 110  J = 1,NODTOT
      FLOG    = 0.0
      FTAN    = PI
      IF (J .EQ. I)      GO TO 100
      DXJ     = XMID - X(J)
      DXJP    = XMID - X(J+1)
      DYJ     = YMID - Y(J)
      DYJP    = YMID - Y(J+1)
      FLOG    = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN    = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
 100  CTIMTJ  = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ  = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      A(I,J)  = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
      B       = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
      A(I,KUTTA) = A(I,KUTTA) + B
      IF ((I .GT. 1) .AND. (I .LT. NODTOT))GO TO 110
C
C           --   IF I-TH PANEL TOUCHES TRAILING EDGE,
C                ADD CONTRIBUTION TO KUTTA CONDITION
C
      A(KUTTA,J) = A(KUTTA,J) - B
      A(KUTTA,KUTTA) = A(KUTTA,KUTTA) + A(I,J)
 110  CONTINUE
C
C           FILL IN KNOWN SIDES
C
      A(I,KUTTA+1) = SINTHE(I)*COSALF - COSTHE(I)*SINALF
 120  CONTINUE
      A(KUTTA,KUTTA+1) = - (COSTHE(1) + COSTHE(NODTOT))*COSALF
     +                   - (SINTHE(1) + SINTHE(NODTOT))*SINALF
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     SUBROUTINE CORVOR (SINALF,COSALF)                               C
C                                                                     C
C           COMPUTE THE LOCAL VELOCITIES OF CORE VORTICES             C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE CORVOR (SINALF,COSALF)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CCVX(200),CCVY(200)
      COMMON /INF3/ AMY(200,201),BMY(200,201)
      COMMON /INF4/ CMY(200,200),CMX(200,200)
      COMMON /POT/ PHI(200),PHIK(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      IF (M .EQ. 1) GO TO 40
      MM1     = M - 1
C
C        VELOCITY COMPONENTS OF CORE VORTICES AT CURRENT TIME STEP
C
      UGC     = 0.0
      VGC     = 0.0
```

```
      DO 10   N = 1,MM1
      XG      = XC(N)*COSALF + YC(N)*SINALF
      IF (XG .GT. XGF) GO TO 5
      UGC     = UGUST
      VGC     = VGUST
 5    VY      = SS*BMY(N,NP1)*(GAMMA-GAMK)/WAKE+
     +            (1.+UGC)*SINALF+VGC*COSALF
      VX      = SS*AMY(N,NP1)*(GAMMA-GAMK)/WAKE+
     +            (1.+UGC)*COSALF-VGC*SINAL
      DO 20   J = 1,NODTOT
      VY      = VY + AMY(N,J)*QK(J) + BMY(N,J)*GAMK
      VX      = VX - BMY(N,J)*QK(J) + AMY(N,J)*GAMK
 20   CONTINUE
C
C        ADD CORE VORTEX CONTRIBUTION
C
      DO 30   MC = 1,MM1
      IF (MC .EQ. N) GO TO 30
      VY      = VY + CMY(N,MC)*CV(MC)
      VX      = VX + CMX(N,MC)*CV(MC)
 30   CONTINUE
C
C        COORDINATES OF CORE VORTICES AT NEXT TIME STEP
C
      CCVX(N) = VX
      CCVY(N) = VY
 10   CONTINUE
 40   CONTINUE
 50   CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     SUBROUTINE FANDM(SINALF,COSALF)                                 C
C                                                                     C
C             COMPUTE AND PRINT OUT CD,CL,CM                          C
C             INTEGRATE PRESSURE DISTRIBUTION BY TRAPEZOIDAL RULE     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE FANDM(SINALF,COSALF)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /CPD/ CP(200)
      CFX     = 0.0
      CFY     = 0.0
      CM      = 0.0
      DO 100  I = 1,NODTOT
      XMID    = .5*(X(I) + X(I+1))
      YMID    = .5*(Y(I) + Y(I+1))
      DX      = X(I+1) - X(I)
      DY      = Y(I+1) - Y(I)
      CFX     = CFX + CP(I)*DY
      CFY     = CFY - CP(I)*DX
      CM      = CM + CP(I)*(DX*XMID + DY*YMID)
 100  CONTINUE
      CD      = CFX*COSALF + CFY*SINALF
      CL      = CFY*COSALF - CFX*SINALF
      WRITE (6,1000) CD,CL,CM
 1000 FORMAT(//,'    CD =',F10.6,'    CL =',F10.6,'    CM =',F10.6)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     SUBROUTINE GAUSS(NRHS)                                          C
C                                                                     C
C        SOLUTION OF LINEAR ALGEBRAIC SYSTEM BY                       C
C        GAUSS ELIMINATION WITH PARTIAL PIVOTING                      C
C                                                                     C
C              A         = COEFFICIENT MATRIX                         C
C             NEQNS      = NUMBER OF EQUATIONS                        C
```

```
C             NRHS      = NUMBER OF RIGHT HAND SIDES                      C
C                                                                         C
C             RIGHT-HAND SIDES AND SOLUTIONS STORED IN                    C
C             COLUMNS NEQNS+1 THRU NEQNS+NRHS OF  A                       C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE GAUSS(NRHS)
      COMMON /COF/ A(201,211),NEQNS
      NP      = NEQNS + 1
      NTOT    = NEQNS + NRHS
C
C             GAUSS REDUCTION
C
      DO 150  I = 2,NEQNS
C
C             --  SEARCH FOR LARGEST ENTRY IN (I-1)TH COLUMN
C                 ON OR BELOW MAIN DIAGONAL
C
      IM      = I - 1
      IMAX    = IM
      AMAX    = ABS(A(IM,IM))
      DO 110  J = I,NEQNS
      IF (AMAX .GE. ABS(A(J,IM))) GO TO 110
      IMAX    = J
      AMAX    = ABS(A(J,IM))
 110  CONTINUE
C
C             --   SWITCH (I-1)TH AND IMAXTH EQUATIONS
C
      IF (IMAX .EQ. IM)   GO TO 140
      DO 130  J = IM,NTOT
      TEMP    = A(IM,J)
      A(IM,J) = A(IMAX,J)
      A(IMAX,J) = TEMP
 130  CONTINUE
C
C             ELIMINATE (I-1)TH UNKNOWN FROM
C             ITH THRU (NEQNS)TH EQUATIONS
C
 140  DO 150  J = I,NEQNS
      R       = A(J,IM)/A(IM,IM)
      DO 150  K = I,NTOT
 150  A(J,K)  = A(J,K) - R*A(IM,K)
C
C             BACK SUBSTITUTION
C
      DO 220  K = NP,NTOT
      A(NEQNS,K) = A(NEQNS,K)/A(NEQNS,NEQNS)
      DO 210  L = 2,NEQNS
      I       = NEQNS + 1 - L
      IP      = I + 1
      DO 200  J = IP,NEQNS
 200  A(I,K)  = A(I,K) - A(I,J)*A(J,K)
 210  A(I,K)  = A(I,K)/A(I,I)
 220  CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C     SUBROUTINE INDATA                                                   C
C                                                                         C
C                  SET PARAMETERS OF BODY SHAPE                           C
C                  FLOW SITUATION, AND NODE DISTRIBUTION                  C
C                                                                         C
C                  USER MUST INPUT                                        C
C                       NLOWER = NUMBER OF NODES ON LOWER SURFACE         C
C                       NUPPER = NUMBER OF NODES ON UPPER SURFACE         C
C                  PLUS DATA ON BODY AND SUBROUTINE BODY                  C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      SUBROUTINE INDATA
      DIMENSION TITLE(20)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
      READ (1,501) ITITLE
      WRITE (6,501) ITITLE
      DO 10   I = 1,ITITLE
      READ (1,502) TITLE
 10   WRITE (6,503) TITLE
 501  FORMAT(3I5)
 502  FORMAT(20A4)
 503  FORMAT(1X,20A4)
      READ (1,501) IFLAG,NLOWER,NUPPER
      WRITE (6,501) IFLAG,NLOWER,NUPPER
      IF (IFLAG .NE. 0) RETURN
      READ (1,501) NACA
      WRITE (6,501) NACA
      IEPS    = NACA/1000
      IPTMAX  = NACA/100 - 10*IEPS
      ITAU    = NACA - 1000*IEPS - 100*IPTMAX
      EPSMAX  = IEPS*0.01
      PTMAX   = IPTMAX*0.1
      TAU     = ITAU*0.01
      IF (IEPS .LT. 10) RETURN
      PTMAX   = 0.2025
      EPSMAX  = 2.6595*PTMAX**3
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C     SUBROUTINE INFL (NITR)                                         C
C                                                                    C
C            CALCULATE INFLUENCE COEFFICIENTS                        C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INFL (NITR)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /NUM/ PI,PI2INV
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CCVX(200),CCVY(200)
      COMMON /INF1/ AAN(201,201),BBN(201,201),AYNP1(201),BYNP1(201)
      COMMON /INF2/ CCN(201,200),CCT(201,200),CYNP1(200),CXNP1(200)
      COMMON /INF3/ AMY(200,201),BMY(200,201)
      COMMON /INF4/ CMY(200,200),CMX(200,200)
      IF ((M .GT. 1) .OR. (NITR .GT. 0)) GO TO 510
C
C     AAN(I,J) : NORMAL VELOCITY INDUCED AT MID-POINT OF I-TH PANEL
C                BY UNIT STRENGTH DISTRIBUTED SOURCE ON THE J-TH PANEL
C
C     BBN(I,J) : NORMAL VELOCITY INDUCED AT MID-POINT OF I-TH PANEL
C                BY UNIT STRENGTH DISTRIBUTED VORTEX ON THE J-TH PANEL
C
      DO 120  I = 1,NODTOT
      XMID    = .5*(X(I) + X(I+1))
      YMID    = .5*(Y(I) + Y(I+1))
      DO 110  J = 1,NODTOT
      FLOG    = 0.0
      FTAN    = PI
      IF (J .EQ. I)     GO TO 100
      DXJ     = XMID - X(J)
      DXJP    = XMID - X(J+1)
      DYJ     = YMID - Y(J)
      DYJP    = YMID - Y(J+1)
      FLOG    = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN    = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
 100  CTIMTJ  = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ  = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      AAN(I,J) = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
```

```
      BBN(I,J) = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
  110 CONTINUE
  120 CONTINUE
  510 CONTINUE
      I        = NP1
      XMID     = .5*(X(I) + X(I+1))
      YMID     = .5*(Y(I) + Y(I+1))
      DO 130  J = 1,NP1
      FLOG     = 0.0
      FTAN     = PI
      IF (J .EQ. I)      GO TO 135
      DXJ      = XMID - X(J)
      DXJP     = XMID - X(J+1)
      DYJ      = YMID - Y(J)
      DYJP     = YMID - Y(J+1)
      FLOG     = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN     = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
  135 CTIMTJ   = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ   = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      AAN(I,J) = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
      BBN(I,J) = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
C
C     AYNP1(J) : Y - VELOCITY INDUCED AT MID POINT OF WAKE ELEMENT
C                (NP1-TH PANEL) BY UNIT STRENGTH DISTRIBUTED SOURCE
C                ON J-TH PANEL
C
C     BYNP1(J) : Y - VELOCITY INDUCED AT MID POINT OF WAKE ELEMENT
C                (NP1-TH PANEL) BY UNIT STRENGTH DISTRIBUTED VORTEX
C                ON J-TH PANEL
C
      AYNP1(J) = PI2INV*(FTAN*COSTHE(J) - FLOG*SINTHE(J))
      BYNP1(J) = PI2INV*(FLOG*COSTHE(J) + FTAN*SINTHE(J))
  130 CONTINUE
      DO 140  I = 1,NODTOT
      XMID     = .5*(X(I) + X(I+1))
      YMID     = .5*(Y(I) + Y(I+1))
      J        = NP1
      DXJ      = XMID - X(J)
      DXJP     = XMID - X(J+1)
      DYJ      = YMID - Y(J)
      DYJP     = YMID - Y(J+1)
      FLOG     = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN     = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
      CTIMTJ   = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ   = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      AAN(I,J) = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
      BBN(I,J) = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
  140 CONTINUE
C
C     CCN(I,J) : NORMAL VELOCITY INDUCED AT MID-POINT OF I-TH PANEL
C                BY UNIT STRENGTH N-TH CORE VORTEX
C
C     CCT(I,J) : TANGENTIAL VELOCITY INDUCED AT MID-POINT OF I-TH PANEL
C                BY UNIT STRENGTH N-TH CORE VORTEX
C
      IF (M .EQ. 1) RETURN
      MM1      = M - 1
      IF (NITR .GT. 0) GO TO 520
      DO 220  I = 1,NODTOT
      XMID     = 0.5*(X(I) + X(I+1))
      YMID     = 0.5*(Y(I) + Y(I+1))
      DO 210  N = 1,MM1
      DX       = XMID - XC(N)
      DY       = YMID - YC(N)
      DIST     = SQRT(DX*DX+DY*DY)
      COSTHN   = DX/DIST
      SINTHN   = DY/DIST
      CTIMTN   = COSTHE(I)*COSTHN + SINTHE(I)*SINTHN
      STIMTN   = SINTHE(I)*COSTHN - COSTHE(I)*SINTHN
      CCN(I,N) = -PI2INV*CTIMTN/DIST
```

```
      CCT(I,N) = -PI2INV*STIMTN/DIST
  210 CONTINUE
  220 CONTINUE
  520 CONTINUE
      I        = NP1
      XMID     = 0.5*(X(I) + X(I+1))
      YMID     = 0.5*(Y(I) + Y(I+1))
      DO 230   N = 1,MM1
      DX       = XMID - XC(N)
      DY       = YMID - YC(N)
      DIST     = SQRT(DX*DX+DY*DY)
      COSTHN   = DX/DIST
      SINTHN   = DY/DIST
      CTIMTN   = COSTHE(I)*COSTHN + SINTHE(I)*SINTHN
      STIMTN   = SINTHE(I)*COSTHN - COSTHE(I)*SINTHN
      CCN(I,N) = -PI2INV*CTIMTN/DIST
      CCT(I,N) = -PI2INV*STIMTN/DIST
C
C     CYNP1(N) : Y - VELOCITY INDUCED AT MID POINT OF WAKE ELEMENT
C                (NP1-TH PANEL) BY UNIT STRENGTH N-TH CORE VORTEX
C
C     CXNP1(N) : X - VELOCITY INDUCED AT MID POINT OF WAKE ELEMENT
C                (NP1-TH PANEL) BY UNIT STRENGTH N-TH CORE VORTEX
C
      CYNP1(N) = -PI2INV*COSTHN/DIST
      CXNP1(N) = +PI2INV*SINTHN/DIST
  230 CONTINUE
C
C     AMY(N,J) : Y - VELOCITY INDUCED AT N-TH CORE VORTEX BY UNIT
C                STRENGTH DISTRIBUTED SOURCE ON THE J-TH PANEL
C
C     BMY(N,J) : Y - VELOCITY INDUCED AT N-TH CORE VORTEX BY UNIT
C                STRENGTH DISTRIBUTED VORTEX ON THE J-TH PANEL
C
      IF (NITR .GT. 0) GO TO 530
      DO 320   N = 1,MM1
      XMID     = XC(N)
      YMID     = YC(N)
      DO 310   J = 1,NODTOT
      DXJ      = XMID - X(J)
      DXJP     = XMID - X(J+1)
      DYJ      = YMID - Y(J)
      DYJP     = YMID - Y(J+1)
      FLOG     = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN     = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
      AMY(N,J) = PI2INV*(FTAN*COSTHE(J) - FLOG*SINTHE(J))
      BMY(N,J) = PI2INV*(FLOG*COSTHE(J) + FTAN*SINTHE(J))
  310 CONTINUE
  320 CONTINUE
  530 CONTINUE
      DO 330   N = 1,MM1
      XMID     = XC(N)
      YMID     = YC(N)
      J        = NP1
      DXJ      = XMID - X(J)
      DXJP     = XMID - X(J+1)
      DYJ      = YMID - Y(J)
      DYJP     = YMID - Y(J+1)
      FLOG     = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN     = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
      AMY(N,J) = PI2INV*(FTAN*COSTHE(J) - FLOG*SINTHE(J))
      BMY(N,J) = PI2INV*(FLOG*COSTHE(J) + FTAN*SINTHE(J))
  330 CONTINUE
C
C     CMY(N,MC) : Y - VELOCITY INDUCED AT N-TH CORE VORTEX BY UNIT
C                 STRENGTH MC-TH CORE VORTEX OTHER THAN ITSELF
C
C     CMX(N,MC) : X - VELOCITY INDUCED AT N-TH CORE VORTEX BY UNIT
C                 STRENGTH MC-TH CORE VORTEX OTHER THAN ITSELF
C
```

```
      IF (NITR .GT. 0) RETURN
      DO 420  N = 1,MM1
      XMID    = XC(N)
      YMID    = YC(N)
      DO 410  MC = 1,MM1
      IF (MC .EQ. N) GO TO 410
      DX      = XMID - XC(MC)
      DY      = YMID - YC(MC)
      DIST    = SQRT(DX*DX+DY*DY)
      COSTHN  = DX/DIST
      SINTHN  = DY/DIST
      CMY(N,MC) = -PI2INV*COSTHN/DIST
      CMX(N,MC) = +PI2INV*SINTHN/DIST
 410  CONTINUE
 420  CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     SUBROUTINE KUTTA (ALPHA,SINALF,COSALF,OMEGA,UX,UY)                C
C                                                                       C
C             USING KUTTA CONDITION TO DETERMINE VORTICITY              C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE KUTTA (ALPHA,SINALF,COSALF,OMEGA,UX,UY)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /COF/ A(201,211),NEQS
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CCVX(200),CCVY(200)
      COMMON /INF1/ AAN(201,201),BBN(201,201),AYNP1(201),BYNP1(201)
      COMMON /INF2/ CCN(201,200),CCT(201,200),CYNP1(200),CXNP1(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      DIMENSION B1(200),B2(200),AA(2),BB(2)
C
C             RETRIEVE SOLUTION FROM A-MATRIX
C
      DO 50   I = 1,NODTOT
      B1(I)   = A(I,NP1)
 50   B2(I)   = A(I,NP2)
C
C             FIND VT AT TRAILING EDGE PANELS
C
      DO 130  K = 1,2
      IF (K .EQ. 1) I = 1
      IF (K .EQ. 2) I = NODTOT
      XMID    = 0.5 * (X(I) + X(I+1))
      YMID    = 0.5 * (Y(I) + Y(I+1))
      VTANG   =   ((1.+UG(I))*COSALF-VG(I)*SINALF+UX)*COSTHE(I)
     +          + ((1.+UG(I))*SINALF+VG(I)*COSALF+UY)*SINTHE(I)
     +          + OMEGA*(YMID*COSTHE(I) - XMID*SINTHE(I))
      AA(K)   = - AAN(I,NP1)*SS/WAKE
      BB(K)   = VTANG + AAN(I,NP1)*SS*GAMMA/WAKE
      DO 120  J = 1,NODTOT
      AA(K)   = AA(K) + AAN(I,J) - BBN(I,J)*B1(J)
      BB(K)   = BB(K) - BBN(I,J)*B2(J)
 120  CONTINUE
C
C          ADD CORE VORTEX CONTRIBUTION
C
      IF (M .EQ. 1) GO TO 100
      MM1     = M - 1
      DO 110  N = 1,MM1
      BB(K)   = BB(K) + CCT(I,N)*CV(N)
 110  CONTINUE
 100  CONTINUE
 130  CONTINUE
C
C             SATISFYING KUTTA CONDITION -- SOLVE FOR VORTEX STRENGTH
```

```
C
      EE      = AA(1)*AA(1) - AA(2)*AA(2)
      FF      = AA(1)*BB(1) - AA(2)*BB(2) - SS/DT
      GG      = BB(1)*BB(1) - BB(2)*BB(2) + 2.*SS*GAMMA/DT
      RADI    = SQRT(FF*FF-EE*GG)
      GAMK    = (-FF - RADI)/EE
C
C             CALCULATE SOURCE STRENGTH
C
      DO 160  I = 1,NODTOT
 160  QK(I)   = GAMK*B1(I) + B2(I)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE NACA45(Z,THICK,CAMBER,BETA)                           C
C                                                                      C
C             EVALUATE THICKNESS AND CAMBER                            C
C             FOR NACA 4- OR 5-DIGIT AIRFOIL                           C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE NACA45(Z,THICK,CAMBER,BETA)
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
      THICK   = 0.0
      IF (Z .LT. 1.E-10)      GO TO 100
      THICK   = 5.*TAU*(.2969*SQRT(Z) - Z*(.126 + Z*(.3537
     +          - Z*(.2843 - Z*.1015))))
 100  IF (EPSMAX .EQ. 0.0) GO TO 130
      IF (NACA .GT. 9999) GO TO 140
      IF (Z .GT. PTMAX) GO TO 110
      CAMBER  = EPSMAX/PTMAX/PTMAX*(2.*PTMAX - Z)*Z
      DCAMDX  = 2.*EPSMAX/PTMAX/PTMAX*(PTMAX - Z)
      GO TO 120
 110  CAMBER  = EPSMAX/(1.-PTMAX)**2*(1. + Z - 2.*PTMAX)*(1. - Z)
      DCAMDX  = 2.*EPSMAX/(1.-PTMAX)**2*(PTMAX - Z)
 120  BETA    = ATAN(DCAMDX)
      RETURN
 130  CAMBER  = 0.0
      BETA    = 0.0
      RETURN
 140  IF (Z .GT. PTMAX)       GO TO 150
      W       = Z/PTMAX
      CAMBER  = EPSMAX*W*((W - 3.)*W + 3. - PTMAX)
      DCAMDX  = EPSMAX*3.*W*(1. - W)/PTMAX
      GO TO 120
 150  CAMBER  = EPSMAX*(1. - Z)
      DCAMDX  = - EPSMAX
      GO TO 120
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE PRESS (SINALF,COSALF,OMEGA,UX,UY)                     C
C                                                                      C
C             COMPUTE UNSTEADY FLOW PRESSURE DISTRIBUTION              C
C                 AND VELOCITY POTENTIAL AT MID-POINTS OF PANELS       C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE PRESS (SINALF,COSALF,OMEGA,UX,UY)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /CPD/ CP(200)
      COMMON /NUM/ PI,PI2INV
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CCVX(200),CCVY(200)
      COMMON /INF1/ AAN(201,201),BBN(201,201),AYNP1(201),BYNP1(201)
      COMMON /INF2/ CCN(201,200),CCT(201,200),CYNP1(200),CXNP1(200)
      COMMON /POT/ PHI(200),PHIK(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      COMMON /EXTV/ UE(200)
```

```
      WRITE (6,1000)
C
C       FIND TANGENTIAL VELOCITY VT AT MID-POINT OF I-TH PANEL
C
      DO 130  I = 1,NODTOT
      XMID    = 0.5 * (X(I) + X(I+1))
      YMID    = 0.5 * (Y(I) + Y(I+1))
      DX      = (X(I+1) - X(I))
      DY      = (Y(I+1) - Y(I))
      DIST    = SQRT(DX*DX+DY*DY)
      VSX     = (1.+UG(I))*COSALF-VG(I)*SINALF + OMEGA*YMID + UX
      VSY     = (1.+UG(I))*SINALF+VG(I)*COSALF - OMEGA*XMID + UY
      VS      = VSX*VSX + VSY*VSY
      VTANG   =   ((1.+UG(I))*COSALF-VG(I)*SINALF+UX)*COSTHE(I)
     +          + ((1.+UG(I))*SINALF+VG(I)*COSALF+UY)*SINTHE(I)
     +          + OMEGA*(YMID*COSTHE(I) - XMID*SINTHE(I))
      VTFREE  = VTANG
      VTANG   = VTANG + SS*(GAMMA-GAMK)*AAN(I,NP1)/WAKE
      DO 120  J = 1,NODTOT
      VTANG   = VTANG - BBN(I,J)*QK(J) + AAN(I,J)*GAMK
 120  CONTINUE
C
C        ADD CORE VORTEX CONTRIBUTION
C
      IF (M .EQ. 1) GO TO 150
      MM1      = M - 1
      DO 140  N = 1,MM1
      VTANG   = VTANG + CCT(I,N)*CV(N)
 140  CONTINUE
 150  CONTINUE
      PHIK(I) = (VTANG-VTFREE)*DIST
      CP(I)   = VS - VTANG*VTANG
      UE(I)   = VTANG
 130  CONTINUE
C
C     COMPUTE DISTURBANCE POTENTIAL BY LINE INTEGRAL OF VELOCITY FIELD
C
C        INTEGRATION FROM UPSTREAM (AT INFINITY) TO THE LEADING EDGE
C
      NPHI    = 10 * NLOWER
      PINK    = 0.0
      XL      = 0.0
      DO 30   L = 1,NPHI
      FRACT   = FLOAT(L)/FLOAT(NPHI)
      XLP     = -10.0 * (1.0 - COS(0.5*PI*FRACT))
      DELX    = XL - XLP
      XMID    = 0.5*(XL+XLP)*COSALF
      YMID    = 0.5*(XL+XLP)*SINALF
      XL      = XLP
      VELX = UGUST
C
C             ADD CONTRIBUTION OF J-TH PANEL
C
      DO 20   J = 1,NP1
      DXJ     = XMID - X(J)
      DXJP    = XMID - X(J+1)
      DYJ     = YMID - Y(J)
      DYJP    = YMID - Y(J+1)
      FLOG    = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN    = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
      CALMTJ  = -COSALF*COSTHE(J) - SINALF*SINTHE(J)
      SALMTJ  = -SINALF*COSTHE(J) + COSALF*SINTHE(J)
      APY     = PI2INV*(FTAN*CALMTJ + FLOG*SALMTJ)
      BPY     = PI2INV*(FLOG*CALMTJ - FTAN*SALMTJ)
      IF (J .EQ. NP1) GO TO 40
      VELX     = VELX  - BPY*QK(J) +GAMK*APY
      GO TO 20
 40   VELX    = VELX  + SS*APY*(GAMMA-GAMK)/WAKE
 20   CONTINUE
C
```

```
C         ADD CORE VORTEX CONTRIBUTION
C
        IF (M .EQ. 1) GO TO 50
        MM1      = M - 1
        DO 60    N = 1,MM1
        DX       = XMID - XC(N)
        DY       = YMID - YC(N)
        DIST     = SQRT(DX*DX+DY*DY)
        COSTHN   = DX/DIST
        SINTHN   = DY/DIST
        SALMTN   = -SINALF*COSTHN + COSALF*SINTHN
        CPT      = -PI2INV*SALMTN/DIST
 60    VELX     = VELX  + CPT*CV(N)
 50    CONTINUE
        PINK     = PINK + VELX * DELX
 30    CONTINUE
C
C       COMPUTE DISTURBANCE POTENTIAL AT MID-POINT OF I-TH PANEL
C
C         LOWER SURFACE
C
        DO 230   I = 1,NLOWER
        PH       = -PINK
        DO 240   J = I,NLOWER
 240   PH       = PH - PHIK(J)
        PHIK(I) = PH
 230   CONTINUE
        DO 270   I = 1,NLOWER-1
        PHIK(I) = 0.5*(PHIK(I) + PHIK(I+1))
 270   CONTINUE
        PHIK(NLOWER) = 0.5*(PHIK(NLOWER) + PINK)
C
C         UPPER SURFACE
C
        DO 250   I = NODTOT,NLOWER+1,-1
        PH       = -PINK
        DO 260   J = NLOWER+1,I
 260   PH       = PH + PHIK(J)
        PHIK(I) = PH
 250   CONTINUE
        DO 280   I = NODTOT,NLOWER+2,-1
        PHIK(I) = 0.5*(PHIK(I) + PHIK(I-1))
 280   CONTINUE
        PHIK(NLOWER+1) = 0.5*(PHIK(NLOWER+1) + PINK)
C
C         COMPUTE CP AT MID POINT OF I-TH PANEL
C
        DO 290   I = 1,NODTOT
        XMID     = .5*(X(I) + X(I+1))
        CP(I)    = CP(I) - 2.*(PHIK(I)-PHI(I))/DT
        WRITE (6,1050) I,XMID,QK(I),GAMK,CP(I),UE(I)
 290   CONTINUE
 1000 FORMAT(/,4X,'J',4X,'X(J)',6X,'Q(J)',5X,'GAMMA',5X,
      + 'CP(J)',6X,'V(J)',/)
 1050 FORMAT(I5,6F10.6)
        RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C      SUBROUTINE SETUP                                                C
C                                                                      C
C              SETUP COORDINATES OF PANEL NODES AND SLOPES OF PANELS   C
C              COORDINATES ARE READ FROM INPUT DATA FILE UNLESS        C
C                  THE AIRFOIL IS OF NACA XXXX OR NACA 230XX TYPE      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE SETUP
        COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
      +        COSTHE(201),SINTHE(201),SS,NP1,NP2
        COMMON /NUM/ PI,PI2INV
```

```
      PI      = 3.1415926585
      PI2INV  = .5/PI
C
C           SET COORDINATES OF NODES ON BODY SURFACE
C
      IF (IFLAG .NE. 0) GO TO 10
      NPOINT  = NLOWER
      SIGN    = -1.0
      NSTART  = 0
      DO 110  NSURF = 1,2
      DO 100  N = 1,NPOINT
      FRACT   = FLOAT(N-1)/FLOAT(NPOINT)
      Z       = .5*(1. - COS(PI*FRACT))
      I       = NSTART + N
      CALL BODY(Z,SIGN,X(I),Y(I))
 100  CONTINUE
      NPOINT  = NUPPER
      SIGN    = 1.0
      NSTART  = NLOWER
 110  CONTINUE
      NODTOT  = NLOWER + NUPPER
      X(NODTOT+1) = X(1)
      Y(NODTOT+1) = Y(1)
      GO TO 20
 10   NODTOT  = NLOWER + NUPPER
      READ (1,501) (X(I),I=1,NODTOT+1)
      WRITE (6,501) (X(I),I=1,NODTOT+1)
      READ (1,501) (Y(I),I=1,NODTOT+1)
      WRITE (6,501) (Y(I),I=1,NODTOT+1)
 501  FORMAT (6F10.6)
 20   NP1     = NODTOT + 1
      NP2     = NODTOT + 2
C
C           SET SLOPES OF PANELS AND CALCULATE AIRFOIL PERIMETER
C
      SS      = 0.0
      DO 200  I = 1,NODTOT
      DX      = X(I+1) - X(I)
      DY      = Y(I+1) - Y(I)
      DIST    = SQRT(DX*DX +DY*DY)
      SS      = SS + DIST
      SINTHE(I) = DY/DIST
      COSTHE(I) = DX/DIST
 200  CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     SUBROUTINE TEWAK (SINALF,COSALF)                                C
C                                                                     C
C           COMPUTE WAKE ELEMENT AT THE TRAILING EDGE                 C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE TEWAK (SINALF,COSALF)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /COF/ A(201,211),NEQS
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /WAK/ VYW,VXW,WAKE,DT
      COMMON /WAK2/ VYWK,VXWK
      COMMON /CORV/ CV(200),XC(200),YC(200),M,TD,CCVX(200),CCVY(200)
      COMMON /INF1/ AAN(201,201),BBN(201,201),AYNP1(201),BYNP1(201)
      COMMON /INF2/ CCN(201,200),CCT(201,200),CYNP1(200),CXNP1(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      XMID    = 0.5 * (X(NP1) + X(NP2))
      YMID    = 0.5 * (Y(NP1) + Y(NP2))
      UGW     = 0.0
      VGW     = 0.0
      XG      = XMID*COSALF + YMID*SINALF
      IF (XG .GT. XGF) GO TO 10
```

```
      UGW     = UGUST
      VGW     = VGUST
 10   VYWK    = (1.+UGW)*SINALF+VGW*COSALF
      VXWK    = (1.+UGW)*COSALF-VGW*SINALF
      DO 120  J = 1,NODTOT
      VYWK    = VYWK + AYNP1(J)*QK(J) + BYNP1(J)*GAMK
 120  VXWK    = VXWK - BYNP1(J)*QK(J) + AYNP1(J)*GAMK
C
C        ADD CORE VORTEX CONTRIBUTION
C
      IF (M .EQ. 1) GO TO 140
      MM1     = M - 1
      DO 130  N = 1,MM1
      VYWK    = VYWK + CYNP1(N)*CV(N)
 130  VXWK    = VXWK + CXNP1(N)*CV(N)
 140  CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE VELDIS(SINALF,COSALF)                                 C
C                                                                      C
C             COMPUTE STEADY FLOW PRESSURE DISTRIBUTION                C
C                 AND VELOCITY POTENTIAL AT MID-POINTS OF PANELS       C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE VELDIS(SINALF,COSALF)
      COMMON /BOD/ IFLAG,NLOWER,NUPPER,NODTOT,X(202),Y(202),
     +             COSTHE(201),SINTHE(201),SS,NP1,NP2
      COMMON /COF/ A(201,211),KUTTA
      COMMON /CPD/ CP(200)
      COMMON /NUM/ PI,PI2INV
      COMMON /SING/ Q(200),GAMMA,QK(200),GAMK
      COMMON /POT/ PHI(200),PHIK(200)
      COMMON /GUST/ UG(200),VG(200),XGF,UGUST,VGUST
      COMMON /EXTV/ UE(200)
      WRITE (6,1000)
C
C             RETRIEVE SOLUTION FROM A-MATRIX
C
      DO 50   I = 1,NODTOT
 50   Q(I)    = A(I,KUTTA+1)
      GAMMA   = A(KUTTA,KUTTA+1)
C
C             FIND VT AND CP AT MID-POINT OF I-TH PANEL
C
      DO 130  I = 1,NODTOT
      XMID    = .5*(X(I) + X(I+1))
      YMID    = .5*(Y(I) + Y(I+1))
      VTANG   = COSALF*COSTHE(I) + SINALF*SINTHE(I)
      VTFREE  = VTANG
C
C             ADD CONTRIBUTION OF J-TH PANEL
C
      DO 120  J = 1,NODTOT
      FLOG    = 0.0
      FTAN    = PI
      IF (J .EQ. I) GO TO 100
      DXJ     = XMID - X(J)
      DXJP    = XMID - X(J+1)
      DYJ     = YMID - Y(J)
      DYJP    = YMID - Y(J+1)
      FLOG    = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN    = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
 100  CTIMTJ  = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ  = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      AA      = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
      B       = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
      VTANG   = VTANG - B*Q(J) +GAMMA*AA
 120  CONTINUE
```

```
      CP(I)    = 1.0 - VTANG*VTANG
      UE(I)    = VTANG
      WRITE (6,1050) I,XMID,Q(I),GAMMA,CP(I),UE(I)
C
C       INITIAL SET-UP FOR DISTURBANCE POTENTIAL CALCULATION
C
      DX       = X(I+1) - X(I)
      DY       = Y(I+1) - Y(I)
      DIST     = SQRT(DX*DX+DY*DY)
      PHI(I)   = (VTANG-VTFREE)*DIST
 130  CONTINUE
C
C     COMPUTE DISTURBANCE POTENTIAL BY LINE INTEGRAL OF VELOCITY FIELD
C
C       INTEGRATION FROM UPSTREAM (AT INFINITY) TO THE LEADING EDGE
C
      NPHI     = 10 * NLOWER
      PIN      = 0.0
      XL       = 0.0
      DO 30    L = 1,NPHI
      FRACT    = FLOAT(L)/FLOAT(NPHI)
      XLP      = -10.0 * (1.0 - COS(0.5*PI*FRACT))
      DELX     = XL - XLP
      XMID     = 0.5*(XL+XLP)*COSALF
      YMID     = 0.5*(XL+XLP)*SINALF
      XL       = XLP
      VELX     =  UGUST
C
C             ADD CONTRIBUTION OF J-TH PANEL
C
      DO 20    J = 1,NODTOT
      DXJ      = XMID - X(J)
      DXJP     = XMID - X(J+1)
      DYJ      = YMID - Y(J)
      DYJP     = YMID - Y(J+1)
      FLOG     = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN     = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
      CALMTJ   = -COSALF*COSTHE(J) - SINALF*SINTHE(J)
      SALMTJ   = -SINALF*COSTHE(J) + COSALF*SINTHE(J)
      APY      = PI2INV*(FTAN*CALMTJ + FLOG*SALMTJ)
      BPY      = PI2INV*(FLOG*CALMTJ - FTAN*SALMTJ)
      VELX     = VELX  - BPY*Q(J) +GAMMA*APY
 20   CONTINUE
      PIN      = PIN + VELX * DELX
 30   CONTINUE
C
C      COMPUTE DISTURBANCE POTENTIAL AT MID-POINT OF I-TH PANEL
C
C       LOWER SURFACE
C
      DO 230   I = 1,NLOWER
      PH       = -PIN
      DO 240   J = I,NLOWER
 240  PH      = PH - PHI(J)
      PHI(I)   = PH
 230  CONTINUE
      DO 270   I = 1,NLOWER-1
      PHI(I)   = 0.5*(PHI(I) + PHI(I+1))
 270  CONTINUE
      PHI(NLOWER) = 0.5*(PHI(NLOWER) + PIN)
C
C       UPPER SURFACE
C
      DO 250   I = NODTOT,NLOWER+1,-1
      PH       = -PIN
      DO 260   J = NLOWER+1,I
 260  PH      = PH + PHI(J)
      PHI(I)   = PH
 250  CONTINUE
      DO 280   I = NODTOT,NLOWER+2,-1
```

```
      PHI(I)  = 0.5*(PHI(I) + PHI(I-1))
280   CONTINUE
      PHI(NLOWER+1) = 0.5*(PHI(NLOWER+1) + PIN)
1000  FORMAT(/,4X,'J',4X,'X(J)',6X,'Q(J)',5X,'GAMMA',5X,
     + 'CP(J)',6X,'V(J)',/)
1050  FORMAT(I5,5F10.6)
      RETURN
      END
```

APPENDIX B

EXAMPLE INPUT DATA FOR PROGRAM U2DIIF

```
 11
*********1*********2*********3*********4*********5*********5*********7
     THIS IS AN EXAMPLE OUTPUT DATA OBTAINABLE FROM PROGRAM U2DIIF
     AIRFOIL : MISES 8.4% THICKNESS (COORDINATES ARE INPUT BY USER)
     PANEL NUMBER : NLOWER = 25 , NUPPER = 25
     AIRFOIL MOTION : MODIFIED RAMP AOA CHANGE ABOUT MID CHORD
     INITIAL AOA : 2.5 DEGREES
     FINAL AOA : 7.5 DEGREES
     AOA RISE TIME : 1.5 CHORD LENGTHS
     COMPUTATION TIME STEP : 0.05 DURING TRANSIENT MOTION, INCREASES
                                  PROGRESSIVELY AFTER FINAL AOA IS REACHED
*********1*********2*********3*********4*********5*********6*********7
  01   25   25
  1.000000  0.994858  0.980866  0.958884  0.929536  0.893455
  0.851308  0.803815  0.751753  0.695948  0.637271  0.576620
  0.514918  0.453098  0.392084  0.332794  0.276105  0.222865
  0.173861  0.129819  0.091393  0.059146  0.033560  0.015010
  0.003767  0.000000  0.003767  0.015008  0.033560  0.059146
  0.091393  0.129819  0.173861  0.222865  0.276105  0.332791
  0.392082  0.453095  0.514915  0.576617  0.637266  0.695946
  0.751750  0.803815  0.851308  0.893455  0.929536  0.958884
  0.980866  0.994858  1.000000
  0.000000 -0.000782 -0.002784 -0.005721 -0.009351 -0.013459
 -0.017837 -0.022285 -0.026618 -0.030671 -0.034289 -0.037341
 -0.039712 -0.041314 -0.042083 -0.041979 -0.040979 -0.039096
 -0.036360 -0.032820 -0.028555 -0.023651 -0.018220 -0.012379
 -0.006259  0.000000  0.006259  0.012379  0.018220  0.023651
  0.028555  0.032820  0.036360  0.039096  0.040979  0.041979
  0.042083  0.041314  0.039712  0.037341  0.034289  0.030671
  0.026618  0.022285  0.017837  0.013459  0.009351  0.005721
  0.002784  0.000782  0.000000
   2.50000  5.000000       1.5       0.0       0.5       0.0      0.00
  0.000000  0.000000  0.000000
  2.000000  0.050000    0.0001  0.000000
```

APPENDIX C
EXAMPLE OUTPUT DATA FROM PROGRAM U2DIIF

```
DATA READ FROM FILE CODE 1

  11
*********1*********2*********3*********4*********5*********5*********7
     THIS IS AN EXAMPLE OUTPUT DATA OBTAINABLE FROM PROGRAM U2DIIF
     AIRFOIL : MISES 8.4% THICKNESS (COORDINATES ARE INPUT BY USER)
     PANEL NUMBER : NLOWER = 25 , NUPPER = 25
     AIRFOIL MOTION : MODIFIED RAMP AOA CHANGE ABOUT MID CHORD
     INITIAL AOA : 2.5 DEGREES
     FINAL AOA : 7.5 DEGREES
     AOA RISE TIME : 1.5 CHORD LENGTHS
     COMPUTATION TIME STEP : 0.05 DURING TRANSIENT MOTION, INCREASES
                     PROGRESSIVELY AFTER FINAL AOA IS REACHED
*********1*********2*********3*********4*********5*********6*********7
   1   25   25
 1.000000  0.994858  0.980866  0.958884  0.929536  0.893455
 0.851308  0.803815  0.751753  0.695948  0.637271  0.576620
 0.514918  0.453098  0.392084  0.332794  0.276105  0.222865
 0.173861  0.129819  0.091393  0.059146  0.033560  0.015010
 0.003767  0.000000  0.003767  0.015008  0.033560  0.059146
 0.091393  0.129819  0.173861  0.222865  0.276105  0.332791
 0.392082  0.453095  0.514915  0.576617  0.637266  0.695946
 0.751750  0.803815  0.851308  0.893455  0.929536  0.958884
 0.980866  0.994858  1.000000
 0.000000 -0.000782 -0.002784 -0.005721 -0.009351 -0.013459
-0.017837 -0.022285 -0.026618 -0.030671 -0.034289 -0.037341
-0.039712 -0.041314 -0.042083 -0.041979 -0.040979 -0.039096
-0.036360 -0.032820 -0.028555 -0.023651 -0.018220 -0.012379
-0.006259  0.000000  0.006259  0.012379  0.018220  0.023651
 0.028555  0.032820  0.036360  0.039096  0.040979  0.041979
 0.042083  0.041314  0.039712  0.037341  0.034289  0.030671
 0.026618  0.022285  0.017837  0.013459  0.009351  0.005721
 0.002784  0.000782  0.000000
 2.500000  5.000000  1.500000  0.000000  0.500000  0.000000  0.000000
 0.000000  0.000000  0.000000
 2.000000  0.050000  0.000100  0.000000

AIRFOIL PERIMETER LENGTH =   2.018599


STEADY FLOW SOLUTION AT ALPHA =   2.500000

   J    X(J)      Q(J)     GAMMA     CP(J)      V(J)

   1  0.997429  0.355723  0.074003  0.316305 -0.826859
   2  0.987862  0.356105  0.074003  0.206074 -0.891025
   3  0.969875  0.365026  0.074003  0.133790 -0.930704
   4  0.944210  0.378836  0.074003  0.082276 -0.957979
   5  0.911495  0.394973  0.074003  0.043033 -0.978247
   6  0.872381  0.412926  0.074003  0.012034 -0.993965
   7  0.827561  0.432568  0.074003 -0.012724 -1.006342
   8  0.777784  0.453710  0.074003 -0.032414 -1.016078
   9  0.723850  0.476112  0.074003 -0.048010 -1.023724
  10  0.666609  0.500047  0.074003 -0.059905 -1.029517
  11  0.606945  0.525455  0.074003 -0.068405 -1.033637
  12  0.545769  0.552654  0.074003 -0.073563 -1.036129
```

13	0.484008	0.581715	0.074003	-0.075309	-1.036971
14	0.422591	0.612882	0.074003	-0.073531	-1.036114
15	0.362439	0.646480	0.074003	-0.067928	-1.033406
16	0.304449	0.683297	0.074003	-0.057668	-1.028430
17	0.249485	0.723595	0.074003	-0.041891	-1.020731
18	0.198363	0.768523	0.074003	-0.019114	-1.009512
19	0.151840	0.819732	0.074003	0.013374	-0.993290
20	0.110606	0.879132	0.074003	0.059669	-0.969707
21	0.075269	0.951277	0.074003	0.127774	-0.933931
22	0.046353	1.043426	0.074003	0.232984	-0.875795
23	0.024285	1.172784	0.074003	0.409236	-0.768612
24	0.009388	1.380641	0.074003	0.727179	-0.522323
25	0.001884	1.653644	0.074003	0.945112	0.234282
26	0.001884	0.545367	0.074003	-0.815326	1.347341
27	0.009387	-0.275210	0.074003	-1.084184	1.443670
28	0.024284	-0.497235	0.074003	-0.872601	1.368430
29	0.046353	-0.580394	0.074003	-0.723499	1.312821
30	0.075269	-0.618032	0.074003	-0.624167	1.274428
31	0.110606	-0.636699	0.074003	-0.552954	1.246176
32	0.151840	-0.645594	0.074003	-0.498371	1.224080
33	0.198363	-0.649755	0.074003	-0.453794	1.205734
34	0.249485	-0.650971	0.074003	-0.415592	1.189787
35	0.304448	-0.650596	0.074003	-0.381557	1.175397
36	0.362436	-0.649624	0.074003	-0.349957	1.161877
37	0.422588	-0.648020	0.074003	-0.319875	1.148858
38	0.484005	-0.646281	0.074003	-0.290579	1.136037
39	0.545766	-0.644572	0.074003	-0.261352	1.123099
40	0.606941	-0.642990	0.074003	-0.231604	1.109776
41	0.666606	-0.641396	0.074003	-0.200941	1.095875
42	0.723848	-0.639980	0.074003	-0.168763	1.081094
43	0.777782	-0.638504	0.074003	-0.134640	1.065195
44	0.827561	-0.637214	0.074003	-0.097676	1.047701
45	0.872381	-0.635880	0.074003	-0.056864	1.028039
46	0.911495	-0.634260	0.074003	-0.010900	1.005436
47	0.944210	-0.632196	0.074003	0.042413	0.978564
48	0.969875	-0.629793	0.074003	0.107609	0.944665
49	0.987862	-0.625960	0.074003	0.193062	0.898297
50	0.997429	-0.619834	0.074003	0.316307	0.826858

CD = 0.000829   CL = 0.303076   CM = -0.080325

```
****************************************
***  BEGIN UNSTEADY FLOW SOLUTION  ****
****************************************
```

TIME STEP TK = 0.050000   TK - TKM1 = 0.050000

ALPHA(T) = 2.516295   OMEGA(T) = -0.011248
U(T) = 0.000000   V(T) = -0.005624

NITR	VXW	VYW	WAKE	THETA	GAMMA
0	0.999048	0.043619	0.050000	0.043633	0.740032E-01
1	0.907832	0.005991	0.045393	0.006600	0.744799E-01
2	0.904138	0.007297	0.045208	0.008070	0.744662E-01
3	0.903985	0.007241	0.045201	0.008010	0.744652E-01

CONVERGED SOLUTION OBTAINED AFTER NITR = 3

J	X(J)	Q(J)	GAMMA	CP(J)	V(J)
1	0.997429	0.435837	0.074466	0.299470	-0.838202

2	0.987862	0.422852	0.074466	0.196120	-0.899569
3	0.969875	0.421798	0.074466	0.132077	-0.936976
4	0.944210	0.427256	0.074466	0.088439	-0.962416
5	0.911495	0.435974	0.074466	0.055859	-0.981181
6	0.872381	0.447181	0.074466	0.029822	-0.995676
7	0.827561	0.460598	0.074466	0.008149	-1.007037
8	0.777784	0.475913	0.074466	-0.010442	-1.015962
9	0.723850	0.492830	0.074466	-0.026773	-1.022942
10	0.666609	0.511571	0.074466	-0.041132	-1.028238
11	0.606945	0.532044	0.074466	-0.053494	-1.032001
12	0.545769	0.554564	0.074466	-0.063581	-1.034262
13	0.484008	0.579201	0.074466	-0.070990	-1.035018
14	0.422591	0.606206	0.074466	-0.075185	-1.034204
15	0.362439	0.635915	0.074466	-0.075487	-1.031672
16	0.304449	0.669133	0.074466	-0.070722	-1.027008
17	0.249485	0.706140	0.074466	-0.059745	-1.019773
18	0.198363	0.748100	0.074466	-0.040826	-1.009171
19	0.151840	0.796683	0.074466	-0.011154	-0.993762
20	0.110606	0.853825	0.074466	0.033396	-0.971231
21	0.075269	0.924099	0.074466	0.100715	-0.936853
22	0.046353	1.014820	0.074466	0.205876	-0.880676
23	0.024235	1.143358	0.074466	0.382654	-0.776542
24	0.009388	1.351849	0.074466	0.703606	-0.535871
25	0.001884	1.634712	0.074466	0.952740	0.209596
26	0.001884	0.564272	0.074466	-0.746972	1.322641
27	0.009387	-0.246433	0.074466	-1.037466	1.430099
28	0.024234	-0.467818	0.074466	-0.838061	1.360476
29	0.046353	-0.551795	0.074466	-0.693563	1.307915
30	0.075269	-0.590860	0.074466	-0.596473	1.271481
31	0.110606	-0.611392	0.074466	-0.527117	1.244626
32	0.151840	-0.622549	0.074466	-0.474818	1.223580
33	0.198363	-0.629333	0.074466	-0.433321	1.206047
34	0.249485	-0.633515	0.074466	-0.399064	1.190716
35	0.304448	-0.636433	0.074466	-0.369826	1.176793
36	0.362436	-0.639059	0.074466	-0.343641	1.163586
37	0.422588	-0.641343	0.074466	-0.319346	1.150745
38	0.484005	-0.643768	0.074466	-0.295850	1.137963
39	0.545766	-0.646483	0.074466	-0.272088	1.124936
40	0.606941	-0.649578	0.074466	-0.247068	1.111386
41	0.666606	-0.652918	0.074466	-0.220048	1.097122
42	0.723848	-0.656699	0.074466	-0.190134	1.081849
43	0.777782	-0.660707	0.074466	-0.156552	1.065285
44	0.827561	-0.665236	0.074466	-0.118313	1.046980
45	0.872381	-0.670135	0.074466	-0.074279	1.026307
46	0.911495	-0.675261	0.074466	-0.023233	1.002476
47	0.944210	-0.680614	0.074466	0.036802	0.974108
48	0.969875	-0.686543	0.074466	0.109850	0.938386
49	0.987862	-0.692719	0.074466	0.203356	0.889792
50	0.997429	-0.699982	0.074466	0.333160	0.815602

CD = 0.001539 CL = 0.302054 CM = -0.088450

TRAILING VORTICES DATA

M	X(M)	Y(M)	CIRC
1	1.022599	0.000131	-0.000933

TIME STEP TK = 0.749999 TK - TKM1 = 0.050000

ALPHA(T) = 4.999996 OMEGA(T) = -0.087266
U(T) = 0.000000 V(T) = -0.043633

NITR	VXW	VYW	WAKE	THETA	GAMMA
0	0.905684	-0.000916	0.045284	-0.001012	0.103235E+00

1 0.905735 -0.000649 0.045287 -0.000717 0.106563E+00

CONVERGED SOLUTION OBTAINED AFTER NITR = 1

J	X(J)	Q(J)	GAMMA	CP(J)	V(J)
1	0.997429	1.115997	0.106565	0.311649	-0.908159
2	0.987862	1.060333	0.106565	0.221106	-0.957033
3	0.969875	1.031653	0.106565	0.170306	-0.983709
4	0.944210	1.013085	0.106565	0.141163	-0.998976
5	0.911495	0.998141	0.106565	0.123949	-1.008098
6	0.872381	0.985265	0.106565	0.114073	-1.013416
7	0.827561	0.974241	0.106565	0.109016	-1.016142
8	0.777784	0.964993	0.106565	0.107136	-1.017013
9	0.723850	0.957451	0.106565	0.107130	-1.016594
10	0.666609	0.952120	0.106565	0.108458	-1.015089
11	0.606945	0.949235	0.106565	0.110657	-1.012669
12	0.545769	0.949393	0.106565	0.113665	-1.009317
13	0.484008	0.952960	0.106565	0.117540	-1.004971
14	0.422591	0.960451	0.106565	0.122480	-0.999505
15	0.362439	0.972455	0.106565	0.128967	-0.992654
16	0.304449	0.990059	0.106565	0.138078	-0.983855
17	0.249485	1.013807	0.106565	0.150962	-0.972486
18	0.198363	1.045081	0.106565	0.169616	-0.957451
19	0.151840	1.085785	0.106565	0.197364	-0.936848
20	0.110606	1.138034	0.106565	0.239070	-0.907675
21	0.075269	1.206453	0.106565	0.303822	-0.863894
22	0.046353	1.298127	0.106565	0.407818	-0.793054
23	0.024285	1.428849	0.106565	0.583190	-0.663132
24	0.009388	1.631804	0.106565	0.872165	-0.369686
25	0.001884	1.819807	0.106565	0.765241	0.485826
26	0.001884	0.373179	0.106565	-1.559307	1.595828
27	0.009387	-0.529374	0.106565	-1.564167	1.590937
28	0.024284	-0.755145	0.106565	-1.202155	1.468068
29	0.046353	-0.836350	0.106565	-0.991312	1.389589
30	0.075269	-0.874103	0.106565	-0.864650	1.338450
31	0.110606	-0.896239	0.106565	-0.781035	1.302194
32	0.151840	-0.912096	0.106565	-0.721021	1.274529
33	0.198363	-0.926607	0.106565	-0.674005	1.251843
34	0.249485	-0.941343	0.106565	-0.634257	1.232132
35	0.304448	-0.957406	0.106565	-0.598321	1.214148
36	0.362436	-0.975544	0.106565	-0.563539	1.196886
37	0.422588	-0.995441	0.106565	-0.528592	1.179831
38	0.484005	-1.017302	0.106565	-0.492391	1.162518
39	0.545766	-1.041010	0.106565	-0.454043	1.144540
40	0.606941	-1.066387	0.106565	-0.412924	1.125555
41	0.666606	-1.093023	0.106565	-0.368740	1.105335
42	0.723848	-1.120818	0.106565	-0.321026	1.083543
43	0.777782	-1.149241	0.106565	-0.269645	1.059939
44	0.827561	-1.178310	0.106565	-0.213999	1.034027
45	0.872381	-1.207644	0.106565	-0.153563	1.005269
46	0.911495	-1.236895	0.106565	-0.087684	0.972965
47	0.944210	-1.265976	0.106565	-0.014749	0.935775
48	0.969875	-1.296105	0.106565	0.069103	0.890819
49	0.987862	-1.330154	0.106565	0.171240	0.832327
50	0.997429	-1.380203	0.106565	0.308161	0.746084

CD = 0.033887 CL = 0.645338 CM = -0.224298

TRAILING VORTICES DATA

M	X(M)	Y(M)	CIRC
1	1.700760	0.077052	-0.000933
2	1.651289	0.072156	-0.001625
3	1.602002	0.066331	-0.002229
4	1.552845	0.060031	-0.002784
5	1.503779	0.053471	-0.003304
6	1.454797	0.046788	-0.003797
7	1.405895	0.040107	-0.004256

8	1.357057	0.033554	-0.004687
9	1.308305	0.027216	-0.005100
10	1.259669	0.021159	-0.005481
11	1.211178	0.015541	-0.005801
12	1.162916	0.010481	-0.006100
13	1.115048	0.006079	-0.006349
14	1.067918	0.002406	-0.006559
15	1.022643	-0.000016	-0.006720

TIME STEP TK = 1.449992 TK - TKM1 = 0.050000

ALPHA(T) = 7.483698 OMEGA(T) = -0.011249
U(T) = 0.000000 V(T) = -0.005625

NITR	VXW	VYW	WAKE	THETA	GAMMA
0	0.901699	0.009872	0.045088	0.010948	0.145377E+00
1	0.901200	0.011028	0.045063	0.012236	0.146997E+00

CONVERGED SOLUTION OBTAINED AFTER NITR = 1

J	X(J)	Q(J)	GAMMA	CP(J)	V(J)
1	0.997429	1.101898	0.146996	0.332408	-0.864078
2	0.987862	1.099609	0.146996	0.228004	-0.921138
3	0.969875	1.116333	0.146996	0.160643	-0.955077
4	0.944210	1.140985	0.146996	0.114868	-0.976579
5	0.911495	1.168619	0.146996	0.082627	-0.990894
6	0.872381	1.197904	0.146996	0.060528	-1.000257
7	0.827561	1.228671	0.146996	0.046710	-1.005878
8	0.777784	1.260826	0.146996	0.039936	-1.008490
9	0.723850	1.294203	0.146996	0.039112	-1.008660
10	0.666609	1.329222	0.146996	0.043766	-1.006560
11	0.606945	1.366027	0.146996	0.053332	-1.002352
12	0.545769	1.405159	0.146996	0.067653	-0.995947
13	0.484008	1.446882	0.146996	0.086578	-0.987214
14	0.422591	1.491619	0.146996	0.110103	-0.975912
15	0.362439	1.539856	0.146996	0.138557	-0.961606
16	0.304449	1.592619	0.146996	0.172964	-0.943442
17	0.249485	1.650390	0.146996	0.214399	-0.920465
18	0.198363	1.714437	0.146996	0.265094	-0.890942
19	0.151840	1.786607	0.146996	0.328746	-0.851951
20	0.110606	1.868882	0.146996	0.410545	-0.798852
21	0.075269	1.965495	0.146996	0.519597	-0.722328
22	0.046353	2.082446	0.146996	0.668455	-0.603478
23	0.024285	2.230950	0.146996	0.866926	-0.394782
24	0.009388	2.419835	0.146996	1.009466	0.050859
25	0.001884	2.339782	0.146996	-0.471799	1.214887
26	0.001884	-0.156346	0.146996	-4.389847	2.320095
27	0.009387	-1.322127	0.146996	-3.033496	2.003043
28	0.024284	-1.560176	0.146996	-2.015200	1.727204
29	0.046353	-1.622657	0.146996	-1.505832	1.569752
30	0.075269	-1.634560	0.146996	-1.212790	1.470549
31	0.110606	-1.628102	0.146996	-1.021804	1.401554
32	0.151840	-1.613626	0.146996	-0.885616	1.350007
33	0.198363	-1.596423	0.146996	-0.780655	1.309004
34	0.249485	-1.578180	0.146996	-0.695048	1.274895
35	0.304448	-1.560042	0.146996	-0.621833	1.245407
36	0.362436	-1.542861	0.146996	-0.556431	1.218916
37	0.422588	-1.526391	0.146996	-0.496616	1.194569
38	0.484005	-1.510876	0.146996	-0.440592	1.171595
39	0.545766	-1.496317	0.146996	-0.387199	1.149437
40	0.606941	-1.482638	0.146996	-0.335611	1.127617
41	0.666606	-1.469497	0.146996	-0.285518	1.105863
42	0.723848	-1.456874	0.146996	-0.236273	1.083745
43	0.777782	-1.444335	0.146996	-0.187542	1.060991

44	0.827561	-1.431955	0.146996	-0.138458	1.037087
45	0.872381	-1.419472	0.146996	-0.088061	1.011468
46	0.911495	-1.406547	0.146996	-0.035055	0.983411
47	0.944210	-1.393024	0.146996	0.023027	0.951546
48	0.969875	-1.379868	0.146996	0.091342	0.912885
49	0.987862	-1.368515	0.146996	0.178655	0.861777
50	0.997429	-1.365097	0.146996	0.303827	0.784388

CD = 0.030956 CL = 0.713821 CM = -0.190685

TRAILING VORTICES DATA

M	X(M)	Y(M)	CIRC
1	2.383780	0.232008	-0.000933
2	2.333846	0.225080	-0.001625
3	2.284404	0.216393	-0.002229
4	2.235273	0.206695	-0.002784
5	2.186347	0.196301	-0.003304
6	2.137600	0.185376	-0.003797
7	2.089004	0.174067	-0.004256
8	2.040442	0.162550	-0.004687
9	1.991957	0.150853	-0.005100
10	1.943572	0.138989	-0.005481
11	1.895155	0.127193	-0.005801
12	1.846658	0.115584	-0.006100
13	1.798126	0.104170	-0.006349
14	1.749496	0.093078	-0.006559
15	1.700793	0.082359	-0.006720
16	1.651988	0.072087	-0.006839
17	1.603077	0.062348	-0.006896
18	1.554079	0.053196	-0.006916
19	1.505019	0.044648	-0.006885
20	1.455917	0.036758	-0.006795
21	1.406791	0.029591	-0.006648
22	1.357685	0.023172	-0.006443
23	1.308651	0.017529	-0.006177
24	1.259734	0.012684	-0.005852
25	1.211021	0.008642	-0.005465
26	1.162620	0.005401	-0.005014
27	1.114706	0.002948	-0.004498
28	1.067624	0.001210	-0.003917
29	1.022530	0.000276	-0.003269

TIME STEP TK = 1.999990 TK - TKM1 = 0.200000

ALPHA(T) = 7.500000 OMEGA(T) = 0.000000
U(T) = 0.000000 V(T) = 0.000000

NITR	VXW	VYW	WAKE	THETA	GAMMA
0	0.939783	0.026143	0.188029	0.027811	0.156187E+00
1	0.948367	0.030228	0.189770	0.031863	0.160749E+00
2	0.948647	0.030081	0.189825	0.031699	0.160765E+00

CONVERGED SOLUTION OBTAINED AFTER NITR = 2

J	X(J)	Q(J)	GAMMA	CP(J)	V(J)
1	0.997429	1.116135	0.160766	0.320772	-0.851401
2	0.987862	1.110748	0.160766	0.221351	-0.907726
3	0.969875	1.126034	0.160766	0.158555	-0.941336
4	0.944210	1.150497	0.160766	0.116479	-0.962935
5	0.911495	1.179177	0.160766	0.086724	-0.977640
6	0.872381	1.210700	0.160766	0.065709	-0.987588
7	0.827561	1.244766	0.160766	0.051593	-0.993866

8	0.777784	1.281128	0.160766	0.043212	-0.997146
9	0.723850	1.319423	0.160766	0.039694	-0.997912
10	0.666609	1.359948	0.160766	0.040813	-0.996296
11	0.606945	1.402723	0.160766	0.046325	-0.992422
12	0.545769	1.448176	0.160766	0.056472	-0.986157
13	0.484008	1.496484	0.160766	0.071440	-0.977368
14	0.422591	1.547997	0.160766	0.091673	-0.965767
15	0.362439	1.603142	0.160766	0.117858	-0.950894
16	0.304449	1.662897	0.160766	0.151404	-0.931843
17	0.249485	1.727721	0.160766	0.193674	-0.907603
18	0.198363	1.798850	0.160766	0.247145	-0.876338
19	0.151840	1.878118	0.160766	0.315652	-0.834967
20	0.110606	1.967481	0.160766	0.404369	-0.778575
21	0.075269	2.071099	0.160766	0.522025	-0.697329
22	0.046353	2.194821	0.160766	0.679683	-0.571292
23	0.024285	2.349161	0.160766	0.881018	-0.350465
24	0.009388	2.539111	0.160766	0.987480	0.119060
25	0.001884	2.420488	0.160766	-0.772858	1.331881
26	0.001884	-0.236994	0.160766	-4.940613	2.437122
27	0.009387	-1.441366	0.160766	-3.294956	2.071290
28	0.024284	-1.678374	0.160766	-2.145366	1.771569
29	0.046353	-1.735029	0.160766	-1.575494	1.601988
30	0.075269	-1.740161	0.160766	-1.248166	1.495590
31	0.110606	-1.726694	0.160766	-1.035354	1.421868
32	0.151840	-1.705142	0.160766	-0.884674	1.367020
33	0.198363	-1.680851	0.160766	-0.770207	1.323621
34	0.249485	-1.655530	0.160766	-0.678837	1.287744
35	0.304448	-1.630341	0.160766	-0.602871	1.256978
36	0.362436	-1.606176	0.160766	-0.537036	1.229567
37	0.422588	-1.582802	0.160766	-0.478581	1.204600
38	0.484005	-1.560513	0.160766	-0.425234	1.181278
39	0.545766	-1.539378	0.160766	-0.375344	1.158993
40	0.606941	-1.519384	0.160766	-0.327563	1.137220
41	0.666606	-1.500277	0.160766	-0.281143	1.115668
42	0.723848	-1.482141	0.160766	-0.235015	1.093871
43	0.777782	-1.464664	0.160766	-0.188536	1.071527
44	0.827561	-1.448072	0.160766	-0.140512	1.048044
45	0.872381	-1.432248	0.160766	-0.089803	1.022803
46	0.911495	-1.417014	0.160766	-0.035089	0.995022
47	0.944210	-1.402392	0.160766	0.026050	0.963238
48	0.969875	-1.389331	0.160766	0.098463	0.924430
49	0.987862	-1.379196	0.160766	0.190332	0.873000
50	0.997429	-1.379105	0.160766	0.319847	0.795184

CD = 0.022037 CL = 0.709635 CM = -0.185327

TRAILING VORTICES DATA

M	X(M)	Y(M)	CIRC
1	2.923984	0.318366	-0.000933
2	2.873369	0.311788	-0.001625
3	2.823534	0.302817	-0.002229
4	2.774192	0.292439	-0.002784
5	2.725182	0.281064	-0.003304
6	2.676492	0.268880	-0.003797
7	2.628086	0.256054	-0.004256
8	2.579755	0.242831	-0.004687
9	2.531597	0.229197	-0.005100
10	2.483748	0.215114	-0.005481
11	2.435886	0.200928	-0.005801
12	2.387918	0.186791	-0.006100
13	2.339993	0.172653	-0.006349
14	2.291922	0.158710	-0.006559
15	2.243772	0.144992	-0.006720
16	2.195477	0.131596	-0.006839
17	2.146968	0.118650	-0.006896
18	2.098253	0.106217	-0.006916
19	2.049394	0.094288	-0.006885
20	2.000362	0.082951	-0.006795

21	1.951119	0.072314	-0.006648
22	1.901708	0.062412	-0.006443
23	1.852145	0.053298	-0.006177
24	1.802427	0.045045	-0.005852
25	1.752599	0.037677	-0.005465
26	1.702674	0.031252	-0.005014
27	1.652678	0.025824	-0.004498
28	1.602603	0.021481	-0.003917
29	1.552431	0.018390	-0.003269
30	1.501920	0.017193	-0.002553
31	1.450588	0.016109	-0.002761
32	1.378677	0.012411	-0.005504
33	1.258439	0.007189	-0.007734
34	1.094864	0.003008	-0.009244

LIST OF REFERENCES

1. Hess, J.L. and Smith, A.M.O., *Calculation of Potential Flow about Arbitrary Bodies*, Progress in Aeronautical Sciences, Vol. 8, Pergamon Press, Oxford, 1966, pp. 1-138.

2. Moran, J., *An Introduction to Theoretical and Computational Aerodynamics*, J. Wiley, New York, 1984, pp. 103-110.

3. Basu, B.C. and Hancock, G.J., *The Unsteady Motion of a Two-Dimensional Aerofoil in Incompressible Inviscid Flow*, Journal of Fluid Mechanics, Vol. 87, Jul 1978, pp. 159-168.

4. Giesing, J.P., *Nonlinear Two-Dimensional Unsteady Potential Flow With Lift*, Journal of Aircraft, Vol. 5, No. 2, Mar-Apr 1968, pp. 135-143.

5. Wagner, H., *Dynamischer Auftrieb von Tragflügeln*, Zeitschrift fuer Angewandte Mathematik und Mechanik, Vol. 5, 1925, p. 17.

6. Ashley, H. and Zartarian, G., *Piston Theory - A New Aerodynamic Tool for the Aeroelastician*, Journal of the Aeronautical Science, Vol. 23, No. 12, Dec 1956.

7. Kim, M.J. and Mook, D.T., *Application of Continuous Vorticity Panels to General Unsteady Incompressible Two-Dimensional Lifting Flows*, Journal of Aircraft, Vol. 23, No. 6, 1986, pp. 464 - 471.

8. Homentcovschi, D., *The Theory of the General Motion of a Thin Profile at a Small Incidence in an Inviscid Incompressible Fluid*, Z. Angew. Math. Mech. Vol. 65, No. 11, 1985, pp. 537 - 544.

9. Halfman, R.L., *Experimental Aerodynamic Derivatives of a Sinusoidally Oscillating Airfoil in Two-Dimensional Flow*, NACA Thirty-Eighth Annual Report, No. 1108, 1952, pp. 1101 - 1144.

10. Küssner, H.G., *Das zweidimensionale Problem der beliebig bewegten Tragfläche unter Berücksichtigung von Partialbewegungen der Flüssigkeit*, Luftfahrtforschung, Vol. 17, 1940, p. 355.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Dept. of Aeronautics, Code 67 Naval Postgraduate School Monterey, California 93943-5000	10
4.	Dynamics Division, Defence Science Organisation Ghim Moh Estate P.O. Box 1050 Singapore 9127 Republic Of Singapore	5
5.	Professor T. Cebeci Staff Director Research & Technology McDonnell Douglas Corporation 3855 Lakewood Boulevard Long Beach, California 90846	1
6.	Mr. Joseph Giesing Senior Staff Engineer Airframe Technology McDonnell Douglas Corporation 3855 Lakewood Boulevard Long Beach, California 90846	1
7.	Professor D.T. Mook Engineering Science and Mechanics Department Virginia Polytechnic Institute and State University Blacksburg, Virginia 24061	1
8.	Mr. Yuan, C.I., SMC 1380 Naval Postgraduate School Monterey, California 93943-5000	1